

Fuzzy autoepistemic logic and its relation to fuzzy answer set programming

Marjon Blondeel^{a,1}, Steven Schockaert^b, Martine De Cock^c, Dirk Vermeir^a

^a*Vrije Universiteit Brussel, Department of Computer Science, Pleinlaan 2, 1050 Brussel, Belgium*

^b*Cardiff University, School of Computer Science and Informatics, 5 The Parade, Cardiff, CF24 3AA, UK*

^c*Ghent University, Department of Applied Mathematics and Computer Science, Krijgslaan 281, 9000 Gent, Belgium*

Abstract

Autoepistemic logic is an important formalism for nonmonotonic reasoning. It extends propositional logic by offering the ability to reason about an agent's (lack of) beliefs. Moreover, it is well known to generalize the stable model semantics of answer set programming. Fuzzy logics on the other hand are multi-valued logics, which allow to model the intensity to which properties are satisfied. We combine these ideas to a fuzzy autoepistemic logic which can be used to reason about one's beliefs in the degrees to which properties are satisfied. We show that many properties from classical autoepistemic logic, e.g. the equivalence between autoepistemic models and stable expansions, remain valid under this generalization. In this paper, we consider a version of fuzzy answer set programming and show that its answer sets can be equivalently described as models in fuzzy autoepistemic logic. We also define a fuzzy logic of minimal belief and negation-as-failure and use this as a tool to show that fuzzy autoepistemic logic generalizes fuzzy answer set programming.

Keywords: Answer set programming, Autoepistemic logic, Fuzzy logics

Email addresses: Marjon.Blondeel@vub.ac.be (Marjon Blondeel), S.Schockaert@cs.cardiff.ac.uk (Steven Schockaert), Martine.DeCock@UGent.be (Martine De Cock), Dirk.Vermeir@vub.ac.be (Dirk Vermeir)

¹Funded by a joint Research Foundation-Flanders (FWO) project

1. Introduction

Answer set programming (ASP) is a tool for modeling combinatorial search problems in a declarative way [1]. Specifically, a search problem is translated into an ASP program, i.e. a set of rules of the form $\alpha \leftarrow \beta$. Such a rule indicates that whenever the body β holds, the head α holds as well. The expression α is a disjunction of literals and β a conjunction of extended literals. A literal is an atom or an expression of the form $\neg a$ with a an atom and \neg classical negation. An extended literal is a literal or an expression of the form $\text{not } l$ with “not” the negation-as-failure operator and l a literal. In ASP we have two types of negation: classical negation and negation-as-failure. The difference between these two is that $\neg a$ is true if we can derive $\neg a$, whereas $\text{not } a$ is true if we fail to derive a . As a result of the occurrence of negation-as-failure, answer set programming is a form of nonmonotonic reasoning. Given an ASP program, the idea is to find a minimal set of literals that can be derived from the program. A program can have several of such “answer sets” or none at all. The answer sets then correspond to the solutions of the original search problem.

Logic programming, which contains ASP as a special case, has had a significant impact on the development of nonmonotonic logics and vice versa [2]. In what follows we will focus on the relationship between ASP and autoepistemic logic, a logic introduced by Moore [3] as a way to reason about one’s own beliefs. Consider for instance my reason for believing that I do not have an older sister. I simply believe that if I had one, I would surely know about it. In autoepistemic logic, one would write “I believe I have a sister.” as the formula $B a$, with B interpreted as “is believed” and a an atom representing the property “I have a sister.” Both ASP and autoepistemic logic are nonmonotonic forms of commonsense reasoning. Autoepistemic logic is nonmonotonic because we can make statements that allow to draw conclusions from a lack of beliefs. For instance, we can express that “If I do not believe P , then Q is true.” Hence, if P cannot be inferred, Q can be derived. On the other hand, if P is added to the set of premises, Q can no longer be derived. Autoepistemic logic is intended to model the beliefs of an ideally rational agent reflecting upon his own beliefs. However, the semantics defined in [3] make it difficult to prove the existence of sets of beliefs satisfying the constraints of autoepistemic logic. Therefore, Moore proposed an alternative possible-world semantics in [4]. If we consider a set of atoms A , truth is defined relative to a pair (I, S) where $S \subseteq \mathcal{P}(A)$ is a

complete S5 structure that represents the agent's beliefs and $I \in \mathcal{P}(A)$ is the actual world. Note that I does not have to be an element of S , because we want to model belief rather than knowledge.

The research oriented towards the relationship between ASP and autoepistemic logic has its roots in the problem of defining semantics for programs with negation-as-failure. Consider for instance the rule $a \leftarrow b \wedge \text{not } c$. Gelfond [5] observed that in autoepistemic logic, this rule can be expressed naturally as the formula $b \wedge \neg(\text{B } c) \rightarrow a$. Gelfond and Lifschitz [6] showed that the answer sets of programs P without classical negation and only one atom in the head of a rule correspond to the models of the autoepistemic logic theory $\lambda(P)$ obtained from P by interpreting rules as material implication and replacing all expressions of the form $\text{not } a$ by $\neg(\text{B } a)$. Unfortunately this translation does not work for programs with more than one atom in the head of rules. For instance the program consisting of the single rule $a \vee b \leftarrow$ would correspond to the autoepistemic formula $a \vee b$ which has exactly one model. The program itself however has two answer sets $\{a\}$ and $\{b\}$. If classical negation is allowed and disjunction in the head is not allowed, problems arise since material implication is contrapositive and rules in ASP are not. This observation led Gelfond and Lifschitz [7] to reject autoepistemic logic as a tool for the study of logical programming, using a semantics based on default logic instead. In [8] however, Lifschitz and Schwarz showed that programs with classical negation and disjunction in the head can easily be represented by autoepistemic theories. In this translation, a rule of the form

$$a_1 \vee \dots \vee a_n \leftarrow b_1 \wedge \dots \wedge b_m \wedge \text{not } c_1 \wedge \dots \wedge \text{not } c_k$$

is transformed into the autoepistemic formula

$$(b_1 \wedge \text{B } b_1) \wedge \dots \wedge (b_m \wedge \text{B } b_m) \wedge \neg(\text{B } c_1) \wedge \dots \wedge \neg(\text{B } c_k) \rightarrow \\ (a_1 \wedge \text{B } a_1) \vee \dots \vee (a_n \wedge \text{B } a_n).$$

The results in [8] are based on the logic of minimal belief and negation as failure (MBNF) ([9], [10]), which uses two independent modal operators corresponding to resp. a “minimal belief” modality and negation-as-failure. This correspondence was independently found by Chen [11], also using MBNF as a starting point. The negation-as-failure modality in MBNF exactly corresponds to negative introspection in autoepistemic logic. MBNF is thus an extension of autoepistemic logic with the “minimal knowledge operator” due

to Halpern and Moses [12]. A simplified version of MBNF (from [10]), which will also be used in this paper, can be used to simulate some forms of default logic and circumscription, as well as some logic programming languages.

Fuzzy answer set programming (FASP) (e.g. [13], [14]), is a generalization of ASP based on fuzzy logics [15], a class of logics whose semantics are based on truth values taken from the unit interval $[0, 1]$. In a FASP program, a rule of the form $\alpha \leftarrow \beta$ intuitively means that the truth degree of α must be greater than or equal to the truth degree of β . By allowing infinitely many truth degrees, FASP can model search problems with continuous domains, entirely similar as ASP does for search problems with discrete domains.

In this paper we combine autoepistemic logic and fuzzy logics. The resulting fuzzy autoepistemic logic is useful to reflect on one's beliefs about the *degrees* to which some properties are satisfied. Consider for example my reason for not believing that my sister smokes *a lot*. If she smoked *a lot*, her breath would smell *often*. Since I do not smell it *often*, I do not believe she smokes *a lot*. Intuitively, if the truth value of $B\varphi$ is equal to c , this means that it is fully believed that φ is true *at least* to degree c . Hence, from one point of view φ is believed to the degree c and from another point of view, there is a Boolean form of belief that the truth value of φ is at least c . Furthermore, note how these views generalize the notion of belief from classical autoepistemic logic, in the sense that having $B\varphi$ false corresponds to having φ true to at least degree 0, i.e. being completely ignorant about φ , and having $B\varphi$ true corresponds to having φ true at least to degree 1, i.e. believing φ to be true. We show that many important properties from classical autoepistemic logic remain valid when generalizing to fuzzy autoepistemic logic. For programs with exactly one atom or constant in the head, we show that the answer sets of a FASP program correspond to the models of an associated fuzzy autoepistemic logic theory. Specifically, it turns out that a straightforward generalization of the translation from ASP to autoepistemic logic works. However, this correspondence is not valid for programs with more complex formulas in the head. To deal with such FASP programs, we will introduce a fuzzy version of MBNF. This will provide us with a tool to show that for these type of programs the answer sets correspond to particular models of an associated fuzzy autoepistemic theory. The fact that this important relationship is preserved provides further insight into the nature of FASP, and at the same time serves as a justification for the particular fuzzy autoepistemic logic we introduce in this paper.

Note that the language of (fuzzy) autoepistemic logic is much more ex-

pressive than the theories we need to represent the (fuzzy) answer set programs. Among others purposes, this could serve as a useful basis to define or compare extensions to the basic language of ASP since the computational complexity does not increase when moving from ASP to autoepistemic logic. Although we have not yet analyzed the computational complexity of fuzzy autoepistemic logic we have no reason to assume that it would be harder than FASP. This might open doors to define extensions for FASP.

This paper is an extension of [16] in which we introduced fuzzy autoepistemic logic and presented some of its properties. The main result of [16] is a method to translate a normal FASP program (cf. Section 3.4) into a fuzzy autoepistemic theory T . In this paper, we will extend this result to more general FASP programs. As we will show by a counterexample, the straightforward translation from [16] cannot directly be extended to this more general setting. Therefore, we propose a new translation which remains valid for more general FASP programs. Furthermore, the proofs for all results have been added in an appendix.

In the next sections we discuss some related work and relevant background on ASP, autoepistemic logic and FASP. In Section 4 we introduce fuzzy autoepistemic logic, investigate some of its properties and in Section 5 we give a motivating example. In Section 6 we analyze the relationship between FASP with singletons in the heads of rules and fuzzy autoepistemic logic. In Section 7, we show that fuzzy autoepistemic logic generalizes a rather general form of FASP. To do so, we will define fuzzy MBNF. A conclusion is given in Section 8.

2. Related work

Epistemic logic. Epistemic logic, the logic of epistemic notions such as knowledge and belief, is a major area of research in artificial intelligence. Von Wright’s seminal work [17] is widely recognized as having initiated the formal study of epistemic logic as we know it today. Since then, various axiomatizations have been proposed, mainly in terms of possible-worlds semantics. An overview is given in [18]. Note that in general, epistemic logics may allow to model the beliefs of several agents, whereas autoepistemic logic is restricted to one’s own beliefs.

Moore’s work about autoepistemic logic [3] was originally based on ideas from McDermott and Doyle ([19], [20]). In [20] an operator M was introduced

which has to be read as “consistent”. Nonmonotonicity was introduced because of the inference rule that “ MP is derivable if $\neg P$ is not derivable.” However, as pointed out in [20], MP is not inconsistent with $\neg P$. In [19] the standard modal logics T, S4 and S5 were extended to nonmonotonic logics, although in the case of S5 it was shown that introducing nonmonotonicity is redundant: the theories of nonmonotonic S5 are exactly those of the standard version of S5. Moore [3] shows why some of the problems McDermott and Doyle encountered arise and how they can be avoided. The language Moore defines is much like McDermott and Doyle’s; a propositional logical language augmented by an epistemic modal operator B interpreted as “is believed”.

Autoepistemic logic is closely related to the work of Halpern and Moses [12], the major difference being that in [12] a logic of knowledge rather than belief is considered. Truth is now defined relative to pairs (I, S) with I an interpretation and S a set of interpretations such that $I \in S$. It then follows that $Bp \rightarrow p$ (with B here interpreted as “is known”) is an axiom. It was obtained independently by Moore and by Halpern and Moses that the sets of formulas that are true in every world of some S5 structure, in which every world is accessible from every world, are exactly the stable autoepistemic theories.

In [21] a recently defined epistemic modal logic can be found. In this paper Meta-Epistemic Logic, which deals with how an agent reasons about what he knows about the beliefs of an other agent, is introduced. Like with autoepistemic logic, the epistemic states are families of non empty subsets of classical interpretations.

Embedding ASP in autoepistemic logic. Before Lifschitz and Schwarz proposed their translation for programs with classical negation and disjunction in the head, other translations of logic programming to autoepistemic logic had been investigated in [22], [23], [24] and [25], but none of them seemed to extend to logic programs with classical negation and disjunction in a natural way. Besides autoepistemic logic and MBNF, reflexive autoepistemic logic has also been used to characterize the semantics of ASP [8]. Reflexive autoepistemic logic [26] has several almost identical semantic characterizations of expansions as autoepistemic logic but defines the modality so that it models knowledge rather than belief. A formula is believed if it is true in all possible worlds w.r.t. the beliefs of the agent and in the actual world. The major difference with autoepistemic logic is that belief allows cyclic arguments and knowledge does not. Reflexive autoepistemic logic turns out

to be very convenient to study answer sets of programs with disjunction in the head and classical negation in front of atoms. The translation of an ASP program to a reflexive autoepistemic theory is even simpler than to an autoepistemic theory.

FASP. In recent years a variety of approaches to FASP have been proposed (e.g. [13], [14], [27], [28], [29], [30], [31], [32], [33]). Although it has been studied by several authors, FASP is by far not as developed as ASP. For example, very little is known about its computational complexity and few techniques are known to compute the answer sets of FASP programs. Also, many extensions proposed for ASP have not yet been considered in FASP. With the exceptions of e.g. [27], [34] and [35], most work is even restricted to FASP programs with exactly one atom in the head. In this paper, we consider a rather general form of FASP programs; the heads of rules are not restricted to single atoms and connectives can in principal be interpreted by arbitrary $[0, 1]^n \rightarrow [0, 1]$ -mappings. The fuzzy equilibrium logic introduced in [34], another generalization of FASP, also allows such constructs. In [34] a correspondence between fuzzy equilibrium logic models and answer sets of FASP programs was shown. Apart from these exceptions and the current manuscript, it appears that little work has been done on nonmonotonic fuzzy logics and their relationship with fuzzy answer set programming.

Note that FASP is not used to deal with uncertainty, but with partial truth. See [36] for a discussion on the difference between these two concepts. To deal with uncertainty, ASP can be extended with possibilistic logic (e.g. [37]) or with probability theory (e.g. [38]).

Fuzzy modal logics. Our work is also related to fuzzy modal logics (e.g. [15], [39], [40]). The semantics of fuzzy autoepistemic logic is closely related to the Kripke-style semantics used in most (fuzzy) modal logics. As will become clear in Section 4, in fuzzy autoepistemic logic there is no accessibility relation. Finite many-valued modal logics with graded accessibility relations have been studied in for example [41] and [42]. In [43] this work is extended to finite many-valued non-monotonic modal logics. In particular autoepistemic logic is generalized by allowing a finite number of truth values and an axiomatic definition is given. In [44], reflexive autoepistemic logic is generalized by allowing a finite set of truth values. All these generalizations use finitely many truth values. In our paper we introduce a continuous generalization for autoepistemic logic.

3. Preliminaries

3.1. Answer set programming (ASP)

Consider a set of atoms A . A *disjunctive ASP program* is a finite set of rules of the form

$$r : a_1 \vee \dots \vee a_n \leftarrow b_1 \wedge \dots \wedge b_m \wedge \text{not } c_1 \wedge \dots \wedge \text{not } c_k,$$

with a_i, b_j, c_l literals (atoms or negated atoms) and/or constants ($\bar{1}$ or $\bar{0}$, i.e. true or false) with $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$ and $l \in \{1, \dots, k\}$. The operator “not” is the *negation-as-failure* operator. Intuitively, $\text{not } a$ is true if there is no proof to support a . We refer to the rule by its label r . The expression $a_1 \vee \dots \vee a_n$ is called the head of r and $b_1 \wedge \dots \wedge b_m \wedge \text{not } c_1 \wedge \dots \wedge \text{not } c_k$ is the body. The set of literals occurring in the program is denoted as \mathcal{L}_P and the set of atoms in the program as \mathcal{B}_P . A rule of the form $\bar{0} \leftarrow a$ is usually written as “ $\leftarrow a$ ” and a rule of the form $a \leftarrow \bar{1}$ as “ $a \leftarrow$ ”. We define some special types of programs. If P contains only rules without negation-as-failure, it is called a *positive* disjunctive ASP program. If P contains no literals of the form $\neg a$ and each rule has exactly one atom in the head, it is called a *normal* ASP program. If P is a normal program that is positive it is called a *simple* ASP program.

A (disjunctive) ASP program can be used to encode a certain search problem. The purpose of the answer set semantics, which we will now recall, is to find solutions to this problem. In what follows we will denote the set of subsets of a set A as $\mathcal{P}(A)$. An *interpretation* I of P is any consistent set in $\mathcal{P}(\mathcal{L}_P)$. A consistent set I of literals is such that for each literal a it is not true that $a \in I$ and $\neg a \in I$. A literal a is true w.r.t. to I , denoted as $I \models a$, iff $a \in I$. This interpretation is extended to formulas and rules as follows:

- $I \models \bar{1}$, $I \not\models \bar{0}$, $I \models \text{not } a$ iff $I \not\models a$
- $I \models (\alpha_1 \wedge \alpha_2)$ iff $I \models \alpha_1$ and $I \models \alpha_2$, $I \models (\alpha_1 \vee \alpha_2)$ iff $I \models \alpha_1$ or $I \models \alpha_2$
- $I \models (\alpha_1 \leftarrow \alpha_2)$ iff $I \not\models \alpha_2$ or $I \models \alpha_1$

An interpretation I is called a *model* of a disjunctive ASP program if $I \models r$ for all rules $r \in P$. A model I is *minimal* if there exists no model J such that $J \subset I$. An interpretation I is called an *answer set* of a positive disjunctive ASP program P if it is a minimal model of P . For simple programs P the minimal model is unique. It equals the least fixpoint of the immediate

consequence operator Π_P [45] which maps interpretations to interpretations and is defined as

$$\Pi_P(I) = \{a \mid (a \leftarrow \beta) \in P \text{ and } \beta \subseteq I\},$$

for an interpretation I . To define the semantics for disjunctive ASP programs containing negation-as-failure, we start from a candidate answer set I and compute the Gelfond-Lifschitz reduct P^I [6]. In particular, to obtain the positive disjunctive ASP program P^I we remove all rules in P that contain expressions of the form $\text{not } a$ with $a \in I$. In the remaining rules in P , we remove all expressions of the form $\text{not } a$. The interpretation I is called an answer set of P iff I is an answer set of P^I . Remark that an answer set I of a program P has to be seen as an epistemic state and not as “truth”. For instance $a \notin I$ does not mean that a is false, but that it is not known that a is true.

Example 1. Consider the normal ASP program P with $\mathcal{B}_P = \{a, b\}$:

$$\begin{aligned} b &\leftarrow \text{not } a \\ a &\leftarrow \text{not } b \end{aligned}$$

The interpretation $M_1 = \{a\}$ is an answer set of P since it is a minimal model of P^{M_1} :

$$a \leftarrow$$

Similarly, the interpretation $M_2 = \{b\}$ is also an answer set of P . One can easily verify that there are no other answer sets.

3.2. Autoepistemic logic

The formulas of autoepistemic logic are built from a set of atoms A , the constants true ($\bar{1}$) and false ($\bar{0}$), the usual connectives and a modal operator B , interpreted as “is believed”. For example, if φ is a formula, then $B\varphi$ indicates that φ is believed. Also, $B(\neg\varphi)$ indicates that $\neg\varphi$ is believed and $\neg(B\varphi)$ that φ is not believed. We write L for the language of all propositional formulas over A and L_B for the extension of L with the operator B . Formulas without the operator B are called *objective* and formulas from L_B are called *unimodal*. An *autoepistemic theory* is a set of unimodal formulas. We define $A' = A \cup \{B\varphi \mid \varphi \in L_B\}$, which is an infinite set, even if A is finite. For technical reasons, we sometimes treat A' as a set of atoms, and consider interpretations $I' \in \mathcal{P}(A')$. Thus for example, expressions of the form $B(a \wedge$

Bb) and $B(Ba)$ are atoms but $a \wedge Bb$ is not. This trick allows us to deal with autoepistemic theories in a purely propositional fashion and soundness and completeness theorems from propositional logic are inherited. For clarity, we will refer to the corresponding propositional language as L' . Moore originally defined the semantics for autoepistemic logic by considering interpretations $I' \in \mathcal{P}(A')$. He defined an interpretation I of an autoepistemic theory T as a subset of A' that conforms to the usual truth recursion for propositional logic and a formula $B\alpha$ is said to be true in I iff $\alpha \in T$. A model of T is an interpretation in which all formulas in T are true.

Autoepistemic logic is intended to model the beliefs of an ideally rational agent reflecting upon his own beliefs. These beliefs should possess some closure conditions; they should include whatever the agent can infer by ordinary logic and by reflecting on what he believes. Formally, an autoepistemic theory T representing the beliefs of such an agent should satisfy the following conditions:

1. If $\alpha_1, \dots, \alpha_n$ are in T and β is a logical consequence w.r.t. L' of $\alpha_1, \dots, \alpha_n$, then β is also in T .
2. If α is in T , then $B\alpha$ is in T .
3. If α is not in T , then $\neg(B\alpha)$ is in T .

Stalnaker [46]² describes the state of belief characterized by such a theory as *stable*: no further conclusions can be drawn by an ideally rational agent in such a state. Theories satisfying these conditions are called *stable autoepistemic theories*. A theory E is called *semantically complete* iff it contains all of its logical consequences and it is *sound* w.r.t. an initial set of premises T iff every interpretation of E that is a model of T is also a model of E . Moore [3] showed that the stable autoepistemic theories are exactly those that are semantically complete. Stability alone gives no soundness results; we do not know what an agent should not believe. However, by imposing syntactical constraints on the theory, the notion of soundness can be captured. For a set of premises T , we look for extensions that make it stable: *stable expansions*.

Definition 1. Consider autoepistemic theories E and T . Then E is a *stable expansion* of T iff

$$E = \text{Cn}(T \cup \{B\varphi \mid \varphi \in E\} \cup \{\neg B\varphi \mid \varphi \notin E\}),$$

²Article based on the unpublished manuscript (1980) to which Moore referred in [3].

where $\text{Cn}(X)$ denotes the set of propositional consequences of X w.r.t. the language L' .

As Moore [3] showed, an autoepistemic theory E is sound w.r.t. an initial set of premises T if E is a stable expansion of T .

Using Definition 1, the following proposition can be shown.

Proposition 1. [47] *Consider an autoepistemic theory T . If all formulas in T are objective, then T has exactly one stable expansion.*

Example 2. *We show that $T = \{\neg(\text{B}a) \rightarrow b, \neg(\text{B}b) \rightarrow a\}$ with $a, b \in A$ has two stable expansions.*

Consider the autoepistemic theory $Q = \{a\}$. Since Q is a set of objective formulas, it has exactly one stable expansion E (Proposition 1):

$$E = \text{Cn}(Q \cup C)$$

with

$$C = \{\text{B}\alpha \mid \alpha \in E\} \cup \{\neg(\text{B}\alpha) \mid \alpha \notin E\}.$$

We will show that E is also a stable expansion of T by proving that

$$\text{Cn}(T \cup C) = \text{Cn}(Q \cup C).$$

First, we prove that $a \in \text{Cn}(T \cup C)$. Since $b \notin E$, we have $\neg(\text{B}b) \in C$ and thus $a \in \text{Cn}(T \cup C)$, hence $\text{Cn}(Q \cup C) \subseteq \text{Cn}(T \cup C)$. Conversely, since $a \in E$, we have $\text{B}a \in C$ and thus $(\neg(\text{B}a) \rightarrow b) \in \text{Cn}(Q \cup C)$. We also have $(\neg(\text{B}b) \rightarrow a) \in \text{Cn}(Q \cup C)$, which implies that $\text{Cn}(T \cup C) \subseteq \text{Cn}(Q \cup C)$.

By symmetry, it follows that the unique stable expansion of $\{b\}$ is also a stable expansion of T .

Moore [4] showed that stable autoepistemic theories can be simply characterized by a Kripke-style possible world semantics. Truth is defined relative to a structure (I, S) with $I \in \mathcal{P}(A)$ representing the actual world and $S \subseteq \mathcal{P}(A)$ representing all worlds considered possible; the beliefs of an agent:

1. $(I, S) \models \bar{1}$ and $(I, S) \not\models \bar{0}$
2. For atoms p , $(I, S) \models p$ iff $p \in I$.
3. For a unimodal formula α , $(I, S) \models \text{B}\alpha$ iff for every $J \in S$ it holds that $(J, S) \models \alpha$.

4. For unimodal formulas φ and ψ , the propositional connectives are handled in the usual way e.g. $(I, S) \models (\varphi \wedge \psi)$ iff $(I, S) \models \varphi$ and $(I, S) \models \psi$.

Intuitively, a unimodal formula α is *believed* if it is true in every interpretation which is considered possible.

Definition 2. Consider an autoepistemic theory T and $S \subseteq \mathcal{P}(A)$. Then S is an autoepistemic model of T iff

$$S = \{I \in \mathcal{P}(A) \mid \forall \varphi \in T : (I, S) \models \varphi\}.$$

In other words, the set of possible worlds w.r.t. the beliefs of the agent is an autoepistemic model of T if it is exactly the set of worlds in which all formulas of T are true.

Definition 3. Consider an autoepistemic theory T and $S \subseteq \mathcal{P}(A)$. Then T is called the autoepistemic theory of S iff

$$T = \{\varphi \in L_B \mid \forall I \in S : (I, S) \models \varphi\} = \{\varphi \in L_B \mid \forall I \in \mathcal{P}(A) : (I, S) \models B \varphi\},$$

We will write $\text{Th}(S)$ to denote this set of formulas.

The set $\text{Th}(S)$ thus contains exactly those formulas that are true in every world that is considered possible w.r.t. the beliefs of an agent.

The following proposition describes the relation between stable expansions and autoepistemic models.

Proposition 2. [4] Consider an autoepistemic theory T . An autoepistemic theory E is a stable expansion of T iff $E = \text{Th}(S)$ for some autoepistemic model S of T .

Example 3. Let us now explicitly compute the stable expansion of theory $Q = \{a\}$ from Example 2. Since there is exactly one stable expansion, we know by Proposition 2 that there is also exactly one autoepistemic model. By definition, S is an autoepistemic model of Q iff

$$S = \{I \in \mathcal{P}(A) \mid (I, S) \models a\} = \{I \in \mathcal{P}(A) \mid a \in I\}.$$

Hence, the unique autoepistemic model of $Q = \{a\}$ is the set of all interpretations that contain a . By Proposition 2, the unique stable expansion of Q is

$$E = \text{Th}(S) = \{\varphi \in L_B \mid \forall I \in \mathcal{P}(A) : a \in I \Rightarrow (I, S) \models \varphi\}.$$

We will now discuss the relationship between answer set programming and autoepistemic logic. Gelfond [5] proposed the following transformation from a normal ASP program P not containing constants to an autoepistemic theory $\lambda(P)$. For each rule $c \leftarrow a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_n$ in P , the unimodal formula $a_1 \wedge \dots \wedge a_m \wedge \neg B b_1 \wedge \dots \wedge \neg B b_n \rightarrow c$ is added to $\lambda(P)$. The following result clarifies the relationship between the answer sets of P and the stable expansions of $\lambda(P)$.

Theorem 1. [6] *Consider a normal ASP program P . M is an answer set of P iff $\lambda(P)$ has a stable expansion E such that $M = E \cap \mathcal{B}_P$.*

Example 4. *Consider the normal ASP program P from Example 1 with $\mathcal{B}_P = \{a, b\}$:*

$$\begin{aligned} b &\leftarrow \text{not } a \\ a &\leftarrow \text{not } b \end{aligned}$$

This program has two answer sets: $M_1 = \{a\}$ and $M_2 = \{b\}$. By Examples 2 and 3, we know that the sets E_1 and E_2 are stable expansions of $\lambda(P)$ with

$$E_1 = \{\varphi \in L_B \mid \forall I \in \mathcal{P}(A) : a \in I \Rightarrow (I, S) \models \varphi\}$$

and

$$E_2 = \{\varphi \in L_B \mid \forall I \in \mathcal{P}(A) : b \in I \Rightarrow (I, S) \models \varphi\}.$$

We find $E_1 \cap \mathcal{B}_P = M_1$ and $E_2 \cap \mathcal{B}_P = M_2$.

3.3. Fuzzy logics

Fuzzy logics [15] are a class of logics whose semantics are based on truth degrees taken from the unit interval $[0, 1]$. We will consider fuzzy logics whose formulas are built from a set of atoms A , constants \bar{c} for each $c \in [0, 1] \cap \mathbb{Q}$ and arbitrary n -ary connectives for each $n \in \mathbb{N}$. A *fuzzy interpretation* is a mapping $I : A \rightarrow [0, 1]$, also called a *fuzzy set* on A . The set of all fuzzy sets on A will be written as $\mathcal{F}(A)$. We can extend this fuzzy interpretation I as follows. Each n -ary connective f is interpreted by a function $\mathbf{f} : [0, 1]^n \rightarrow [0, 1]$. We define $[f(\alpha_1, \dots, \alpha_n)]_I = \mathbf{f}([\alpha_1]_I, \dots, [\alpha_n]_I)$ for formulas α_i ($1 \leq i \leq n$). For $c \in [0, 1] \cap \mathbb{Q}$ we have $[\bar{c}]_I = c$. If C is a set of formulas we say that I is a *fuzzy model* of C iff $[\alpha]_I = 1$ for all $\alpha \in C$; we write this as $I \models C$. For fuzzy interpretations $I_1, I_2 \in \mathcal{F}(A)$ we write $I_1 \leq I_2$ iff $I_1(a) \leq I_2(a)$ for all $a \in A$. A model I is a *minimal fuzzy model* of a set of formulas T if there does not exist a fuzzy model J of T such that $J < I$.

We will now recall some generalizations of the classical connectives. In examples, we will often use these semantics, although all theorems will be valid for arbitrary connectives and their interpretations. Specifically, *triangular norms* (short t-norm) are used to generalize the classical conjunction. These are mappings $T : [0, 1]^2 \rightarrow [0, 1]$ that are commutative, associative, increasing and for which it holds that $T(x, 1) = x$ for each $x \in [0, 1]$. Disjunction can be generalized by a *triangular conorm* (short t-conorm). These are mappings $S : [0, 1]^2 \rightarrow [0, 1]$ that are commutative, associative, increasing and for which holds that $S(x, 0) = x$ for each $x \in [0, 1]$. Logical implication can be generalized by an *implicator*, i.e. a function $I : [0, 1]^2 \rightarrow [0, 1]$ such that $I(0, 0) = I(0, 1) = I(1, 1) = 1$ and $I(1, 0) = 0$ and I is decreasing in the first component and increasing in the second. Given a t-norm T , the *residual implicator* I_T of T , defined as

$$I_T(x, y) = \sup\{\lambda \mid \lambda \in [0, 1] \text{ and } T(x, \lambda) \leq y\}$$

satisfies all these conditions. For left-continuous t-norms we have the following property:

Proposition 3. [15] *If T is a left-continuous t-norm, then for all $x, y \in [0, 1]$ it holds that*

$$x \leq y \text{ iff } I_T(x, y) = 1.$$

Consider a residual implicator I and a t-norm T . The *biresiduum* of I and T is the mapping $E_{T,I} : [0, 1]^2 \rightarrow [0, 1]$:

$$E_{T,I}(x, y) = T(I(x, y), I(y, x)).$$

This function is a generalization of the logical equivalence. Note that I does not need to be the residual implicator of T . By Proposition 3 and the definition of a t-norm, we can infer that $E_{T,I}(x, y) = 1$ iff $x = y$. Finally, negation can be generalized by a *negator*. A *negator* is a function $N : [0, 1] \rightarrow [0, 1]$ such that N is decreasing, $N(1) = 0$ and $N(0) = 1$. Every implicator I induces a negator N_I defined as $N_I(x) = I(x, 0)$.

Logics whose semantics are based on triangular norms form an important subclass of fuzzy logics; they generalize the classical logical connectives in a natural way. In examples we will often use the fuzzy logic based on the Łukasiewicz t-norm. For the connectives conjunction \otimes , disjunction \oplus , implication \rightarrow and negation \neg and a fuzzy interpretation $I \in \mathcal{F}(A)$ we have

1. $[\alpha \otimes \beta]_I = \max([\alpha]_I + [\beta]_I - 1, 0)$
2. $[\alpha \oplus \beta]_I = \min([\alpha]_I + [\beta]_I, 1)$
3. $[\alpha \rightarrow \beta]_I = \min(1 - [\alpha]_I + [\beta]_I, 1)$
4. $[\neg \alpha]_I = 1 - [\alpha]_I$

We will also use the Rescher implicator which is defined as

$$[\alpha \rightarrow_R \beta]_I = \begin{cases} 1 & \text{if } [\alpha]_I \leq [\beta]_I \\ 0 & \text{otherwise} \end{cases}$$

Lukasiewicz logic is often used in applications because it preserves many nice properties from classical logic. Moreover, among the t-norm based logics, Lukasiewicz logic is the only one with a continuous implicator. This means that a set of formulas in Lukasiewicz logic can be seen a set of constraints on continuous functions. This logic is also closely related to mixed integer programming. McNaughton [48] showed this in a non constructive way and Hähnle [49] gave a concrete translation from a set of formulas in Lukasiewicz logic into a mixed integer program.

3.4. Fuzzy Answer Set Programming (FASP)

Consider a set of atoms A . A (*general*) *FASP program* is a finite set of rules of the form

$$r : g(a_1, \dots, a_n) \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k),$$

with a_i, b_j, c_l atoms or constants \bar{c} with $c \in ([0, 1] \cap \mathbb{Q})$ ($i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$ and $l \in \{1, \dots, k\}$), \leftarrow an implication and f and g resp. $(m + k)$ -ary and n -ary connectives. For each n -ary function h (for each $n \in \mathbb{N}$) that is occurring in the head or the body of a rule, we assume there is a corresponding connective $\mathbf{h} : [0, 1]^n \rightarrow [0, 1]$ that is increasing in all its arguments. Typically these functions will correspond to the connectives from a given fuzzy logic (Section 3.3), but other choices, e.g. averaging operators, can be useful as well. For \leftarrow , we assume there is a residual implicator. Each negation-as-failure operator not_j is associated with a negator \mathbf{N}_j . We refer to the rule by its label r . The expression $g(a_1, \dots, a_n)$ is called the head r_h of r and $f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k)$ is the body r_b . As for ASP (Section 3.1), we consider some special types of programs. Programs without negation-as-failure are called *positive*, programs with exactly one atom in the head are called *normal* and normal programs that are positive are called *simple*. A

fuzzy interpretation I of a general FASP program P is an element of $\mathcal{F}(\mathcal{B}_P)$, with \mathcal{B}_P the set of atoms occurring in P . We can extend this mapping as follows: $[\bar{c}]_I = c$ for $c \in [0, 1] \cap \mathbb{Q}$, $[a]_I = I(a)$ and $[\text{not}_j a]_I = \mathbb{N}_j([a]_I)$ for $a \in \mathcal{B}_P$ and $[r]_I = \mathbb{I}([r_b]_I, [r_h]_I)$, for a rule $r : r_h \leftarrow r_b$ and a corresponding implicator \mathbb{I} . For fuzzy interpretations I_1 and I_2 we write $I_1 \leq I_2$ iff $I_1(a) \leq I_2(a)$ for all $a \in \mathcal{B}_P$. A fuzzy interpretation I is called a *fuzzy model* of P iff $[r]_I = 1$ for all $r \in P$. A model I is a *minimal fuzzy model* of P if there does not exist a fuzzy model J of P such that $J < I$.

A fuzzy interpretation I is an *answer set* of a positive FASP program P iff it is a minimal fuzzy model of P . Remark that a positive FASP program can have none, one or several answer sets [35]. If P is a simple program, the answer set is unique and it coincides with the least fixpoint of the immediate consequence operator Π_P [28]. This operator maps fuzzy interpretations to fuzzy interpretations and is defined as

$$\Pi_P(I)(a) = \sup\{[r_b]_I \mid (a \leftarrow r_b) \in P\},$$

for $a \in \mathcal{B}_P$ and I a fuzzy interpretation. For programs that are not positive, a generalization of the Gelfond-Lifschitz reduct is used [13]. More precisely, let P be a general FASP program and I a fuzzy interpretation. The reduct of P w.r.t. I is obtained by replacing in each rule r all expressions of the form $\text{not}_j a$ by the interpretation $[\text{not}_j a]_I$, obtaining a new rule r^I . The reduct $P^I = \{r^I \mid r \in P\}$ is then a positive FASP program and I is called an answer set of P iff I is an answer set of P^I .

Example 5. Consider the normal FASP program P with $\mathcal{B}_P = \{a, b\}$:

$$\begin{aligned} b &\leftarrow \text{not } a \\ a &\leftarrow \text{not } b \end{aligned}$$

We assume that \leftarrow and not correspond to resp. the Lukasiewicz implicator and the Lukasiewicz negator and show that for each $x \in [0, 1]$, M_x with $M_x(a) = x$ and $M_x(b) = 1 - x$ is an answer set of P . We first compute the reduct P^{M_x} :

$$\begin{aligned} b &\leftarrow \overline{1 - x} \\ a &\leftarrow \bar{x} \end{aligned}$$

The minimal model of P^{M_x} is then exactly M_x . Note that there are infinitely many answer sets.

Remark 1. *Note that a reduct of a FASP program can contain irrational constants, but this does not cause any problems to the main reasoning tasks for FASP. Indeed, from the analysis of the geometrical structure underlying fuzzy equilibrium models [34], it follows that the geometrical space of all answer sets of a program can be described in terms of polyhedra for which all vertices have rational coordinates. Hence, there is no problem for the “interesting” answer sets. However, semantically one wants to have a continuum of answer sets and thus conceptually, a more general language needs to be used for definitions in terms of reducts.*

4. Fuzzy autoepistemic logic

In this section, we combine the ideas of autoepistemic logic (Section 3.2) and fuzzy logics (Section 3.3). This will provide us with a tool to reason about one’s beliefs in the degree to which one or more properties are satisfied.

The formulas in fuzzy autoepistemic logic are built from a set of atoms A , constants \bar{c} for each $c \in [0, 1] \cap \mathbb{Q}$, the set $F = \cup_{n \in \mathbb{N}} F_n$ with F_n the set of n -ary connectives f and a modal operator B , interpreted as “is believed”. We will denote this language as \widetilde{L}_B . As for classical autoepistemic logic, we will make the distinction between objective and unimodal formulas. A *fuzzy autoepistemic theory* in \widetilde{L}_B is a set of formulas in \widetilde{L}_B . Like in classical autoepistemic logic, beliefs of an agent can be represented by fuzzy autoepistemic theories. As before, we define $A' = A \cup \{B\varphi \mid \varphi \in \widetilde{L}_B\}$.

We define a generalization of stable expansions (Definition 1). Recall that in Definition 1, a classical stable expansion E of an autoepistemic theory T is such that the set of classical models of $T \cup \{B\varphi \mid \varphi \in E\} \cup \{\neg B\varphi \mid \varphi \notin E\}$ is exactly the set of classical models of E . Note that if I is a consistent classical model of $\{B\varphi \mid \varphi \in E\} \cup \{\neg B\varphi \mid \varphi \notin E\}$ that this means that $B\varphi \in I$ iff $\varphi \in E$. For fuzzy sets I' and E , this can be generalized by demanding that $[B\varphi]_{I'} = E(\varphi)$.

Definition 4. *Consider a fuzzy autoepistemic theory T in \widetilde{L}_B and $E \in \mathcal{F}(\widetilde{L}_B)$. We call E a fuzzy stable expansion of T if for each $\alpha \in \widetilde{L}_B$*

$$E(\alpha) = \inf \left\{ [\alpha]_{I'} \mid I' \models T, [B\varphi]_{I'} = E(\varphi) \text{ for each } \varphi \in \widetilde{L}_B \text{ and } I' \in \mathcal{F}(A') \right\}.$$

Remark 2. *Suppose E is a fuzzy stable expansion of $\{\alpha\}$ with $\alpha \in \widetilde{L}_B$. By definition of E we thus have that $[B\alpha]_{I'} = E(\alpha) \leq [\alpha]_{J'}$ for all $J' \in$*

$\mathcal{F}(A')$ such that $J' \models T$ and $[B\varphi]_{J'} = E(\varphi)$ for each $\varphi \in \widetilde{L}_B$. Thus $[B\alpha]_{I'}$ determines a lower bound on the truth degree of α .

We will now generalize Definitions 2 and 3. First, we need to define the satisfaction relation for unimodal formulas. Consider $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$:

1. For atoms p , $[p]_{(I,S)} = I(p)$.
2. For constants \bar{c} , $[\bar{c}]_{(I,S)} = c$.
3. For a unimodal formula α : $[B\alpha]_{(I,S)} = \inf_{J \in S} [\alpha]_{(J,S)}$.
4. For unimodal formulas α_i ($1 \leq i \leq n$) and $f \in F_n$ ($n \in \mathbb{N}$) which is interpreted by $\mathbf{f} : [0, 1]^n \rightarrow [0, 1]$:
 $[f(\alpha_1, \dots, \alpha_n)]_{(I,S)} = \mathbf{f}([\alpha_1]_{(I,S)}, \dots, [\alpha_n]_{(I,S)})$.

Remark 3. Note that the following are equivalent expressions for a formula α , $c \in [0, 1] \cap \mathbb{Q}$ and $S \subseteq \mathcal{F}(A)$:

$$\begin{aligned}
\forall I \in S : [B\alpha]_{(I,S)} \geq c &\Leftrightarrow \forall I \in S : \inf_{J \in S} [\alpha]_{(J,S)} \geq c \\
&\Leftrightarrow \forall J \in S : [\alpha]_{(J,S)} \geq c \\
&\Leftrightarrow \forall J \in S : [\bar{c} \rightarrow \alpha]_{(J,S)} = 1 \\
&\Leftrightarrow \forall I \in \mathcal{F}(A) : [B(\bar{c} \rightarrow \alpha)]_{(I,S)} = 1
\end{aligned}$$

Hence on the one hand, c reflects the degree of belief on α and on the other hand there is a Boolean form of belief on the formula $\bar{c} \rightarrow \alpha$.

Remark 4. In possibilistic logic, the semantics are defined in terms of a possibility distribution over propositional interpretations, i.e. by mappings $\pi : \mathcal{P}(A) \rightarrow [0, 1]$. Such a mapping encodes for each interpretation or possible world I to what extent it is possible that it refers to the real world, or in other words, to what extent available knowledge does not exclude I from being the real world.

Syntactically, a formula in possibilistic logic corresponds to a propositional formula, encapsulated by a graded modality. In particular formulas are of the form (α, λ) , with α a formula in classical propositional logic and $\lambda \in [0, 1]$, with the intended meaning that $\sup \{ \pi(I) \mid I \not\models \alpha \} \leq 1 - \lambda$.

Hence, there is a clear duality between the semantics of fuzzy autoepistemic logic as we have defined it here and the semantics of possibilistic logic. Indeed, whereas we have defined the semantics of fuzzy autoepistemic logic in terms of a classical set of fuzzy interpretations, possibilistic logic is defined in terms of a fuzzy set of classical interpretations. This duality also reflects

the different way in which the modality should be interpreted. In possibilistic logic, and in a number of graded modal logics, the strength by which an agent believes a proposition can be expressed. Degrees of belief are then used to express that some propositions are considered to be more plausible than others. In contrast, our approach does not deal with such strengths of belief; believing a proposition α to degree λ is interpreted as a Boolean belief in the proposition $\lambda \rightarrow \alpha$, i.e. degrees of belief are used to express that some propositions are true to a greater extent than others. Of course, one can also imagine a logic based on fuzzy sets of fuzzy interpretations, as was proposed for example in [50].

Definition 5. Consider a fuzzy autoepistemic theory T in \widetilde{L}_B and $S \subseteq \mathcal{F}(A)$. We call S a fuzzy autoepistemic model of T if

$$S = \{I \in \mathcal{F}(A) \mid \forall \varphi \in T : [\varphi]_{(I,S)} = 1\}.$$

Similar as for classical autoepistemic logic, the set of possible worlds w.r.t. the beliefs of the agent is a fuzzy autoepistemic model of T if it is exactly the set of worlds in which T is true.

Example 6. Suppose $T = \{\neg(Ba) \rightarrow b, \neg(Bb) \rightarrow a\}$ with $a, b \in A$ and the negation and implication from Łukasiewicz logic. Consider $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$. For the first formula of T we have

$$\begin{aligned} [\neg(Ba) \rightarrow b]_{(I,S)} = 1 &\Leftrightarrow [\neg(Ba)]_{(I,S)} \leq [b]_{(I,S)} \\ &\Leftrightarrow 1 - [Ba]_{(I,S)} \leq I(b) \\ &\Leftrightarrow 1 - \inf_{J \in S} J(a) \leq I(b) \\ &\Leftrightarrow 1 - I(b) \leq \inf_{J \in S} J(a) \end{aligned}$$

By symmetry we also have $[\neg(Bb) \rightarrow a]_{(I,S)} = 1 \Leftrightarrow 1 - I(a) \leq \inf_{J \in S} J(b)$. It follows that if S is a fuzzy autoepistemic model of T , then

$$S = \left\{ I \in \mathcal{F}(A) \mid 1 - I(b) \leq \inf_{J \in S} J(a) \text{ and } 1 - I(a) \leq \inf_{J \in S} J(b) \right\}.$$

For such a set S , let $x = \inf_{J \in S} J(a)$ and $y = \inf_{J \in S} J(b)$. We prove that $y = 1 - x$. For each $J \in S$, we have $1 - J(b) \leq x$ and thus

$$1 - y = 1 - \inf_{J \in S} J(b) = \sup_{J \in S} (1 - J(b)) \leq x.$$

Hence $x + y \geq 1$. S contains all fuzzy interpretations I such that $1 - I(b) \leq x$ and $1 - I(a) \leq y$, thus there is certainly a fuzzy interpretation $I_0 \in S$ with $I_0(a) = 1 - y$. We obtain $x = \inf_{J \in S} J(a) \leq I_0(a) = 1 - y$, hence $x + y \leq 1$. From $x + y \leq 1$ and $x + y \geq 1$, it follows that $x + y = 1$. Thus, if S is a fuzzy autoepistemic model of T , then

$$S = \{I \in \mathcal{F}(A) \mid I(b) \geq 1 - x \text{ and } I(a) \geq x\}$$

for some $x \in [0, 1]$. Moreover, we can prove that each set of this form is a fuzzy autoepistemic model of T , such that we can conclude that the fuzzy autoepistemic models of T are exactly all sets of this form. Define for each $x \in [0, 1]$

$$S_x = \{I \in \mathcal{F}(A) \mid I(b) \geq 1 - x \text{ and } I(a) \geq x\}.$$

To conclude that S_x is a fuzzy autoepistemic model of T , we consider $I \in S_x$ and observe that for all $J \in S_x$ we have $1 - I(b) \leq x \leq J(a)$, thus $1 - I(b) \leq \inf_{J \in S_x} J(a)$. Similarly, $1 - I(a) \leq \inf_{J \in S_x} J(b)$.

Definition 6. Consider $S \subseteq \mathcal{F}(A)$. The fuzzy autoepistemic theory of S is the fuzzy set $\text{Th}(S)$ on \widetilde{L}_B such that for each unimodal formula φ

$$\text{Th}(S)(\varphi) = \inf_{I \in S} [\varphi]_{(I, S)} = [\mathbf{B} \varphi]_{(J, S)},$$

with $J \in \mathcal{F}(A)$ arbitrary.

We now present some lemmas that will help us to prove generalizations of Propositions 1 and 2 in respectively Proposition 5 and 4. To prove Proposition 4 we will use the result from Proposition 5. For proofs of the lemmas and propositions we refer to the appendix. For a fuzzy set $I' : A' \rightarrow [0, 1]$, we define $I'_{|A}$ as the fuzzy set $I'_{|A} : A \rightarrow [0, 1] : a \mapsto I'(a)$.

Lemma 1. Let $Q \subseteq \mathcal{F}(A')$ such that for each $I' \in Q$ and $\varphi \in \widetilde{L}_B$ we have that $[\mathbf{B} \varphi]_{I'} = \inf_{J' \in Q} [\varphi]_{J'}$. Then, for $\alpha \in \widetilde{L}_B$, $S = \{I'_{|A} \mid I' \in Q\}$ and $I' \in Q$ we have

$$[\alpha]_{(I'_{|A}, S)} = [\alpha]_{I'}.$$

Lemma 2. Let $S \subseteq \mathcal{F}(A)$ and $I' \in \mathcal{F}(A')$ such that for each $\varphi \in \widetilde{L}_B$ we have $[\mathbf{B} \varphi]_{I'} = \text{Th}(S)(\varphi)$. Then, for $\alpha \in \widetilde{L}_B$ we have

$$[\alpha]_{(I'_{|A}, S)} = [\alpha]_{I'}.$$

Proposition 4. *Consider a fuzzy autoepistemic theory T in \widetilde{L}_B .*

1. *If $E \in \mathcal{F}(\widetilde{L}_B)$ is a fuzzy stable expansion of T , then $E = \text{Th}(S)$ with S a fuzzy autoepistemic model of T .*
2. *If $S \subseteq \mathcal{F}(A)$ is a fuzzy autoepistemic model of T , then $E = \text{Th}(S)$ is a fuzzy stable expansion of T .*

Example 7. *Reconsider the fuzzy autoepistemic theory*

$$T = \{\neg(B a) \rightarrow b, \neg(B b) \rightarrow a\}$$

from Example 6. All fuzzy autoepistemic models are of the form

$$S_x = \{I \in \mathcal{F}(A) \mid I(b) \geq 1 - x \text{ and } I(a) \geq x\},$$

with $x \in [0, 1]$. Hence, all fuzzy stable expansions of T are of the form E_x with $x \in [0, 1]$ defined by $E_x(a) = \text{Th}(S)(a) = \inf_{I \in S} I(a) = x$ and $E_x(b) = \text{Th}(S)(b) = \inf_{I \in S} I(b) = 1 - x$.

Proposition 5. *Every set of objective formulas in \widetilde{L}_B has a unique fuzzy stable expansion.*

Remark 5. *Suppose T is an objective fuzzy autoepistemic theory, such that T contains α and $\neg\alpha$ with α an objective formula and \neg interpreted by a negator. In such a case, the theory T does not have any fuzzy models, but it still has the empty set as its unique fuzzy autoepistemic model.*

5. Motivating example

In this section we will demonstrate how fuzzy autoepistemic logic as defined in Section 4 can be used in a real world scenario.

Forest fires cause massive loss of vegetation and animal life. If a fire is detected on time, suppression units are able to reach the fire in its initial stages which is important to avoid huge losses. Moreover suppression costs will be considerably reduced. Wireless sensor networks can be effectively used for this purpose [51]. These networks consist of a number of devices that can sense their environment and communicate wirelessly. We will use fuzzy autoepistemic logic to determine, given measurements made by the sensors about the temperature, if there are sensors that are not working optimally and if so, within what range we can reasonably assume the temperature to be.

Suppose we have n sensors. For each sensor i ($i \in \{1, \dots, n\}$), we denote the exact temperature at its location as t_i and the measured temperature as t'_i . By assuming an appropriate linear rescaling, we can see temperature as a value in $[0, 1]$. The sensor network defines a weighted graph G as follows. The vertices are the sensors and there is an edge with weight $w_{ij} \in [0, 1]$ between the vertices corresponding to sensor i and sensor j . The value w_{ij} is such that we can reasonably assume, based on the locations of sensors i and j that the temperature difference between these locations must be less than w_{ij} . We will now present two strategies to determine bounds on the actual temperatures, given the values w_{ij} and t'_i ($i, j \in \{1, \dots, n\}$). We will use the connectives from Łukasiewicz logic and the Rescher implicator \rightarrow_R .

Recall that (classical) autoepistemic logic originally was intended to model the beliefs of an agent. These beliefs are represented by a set of formulas; autoepistemic theories. The formulas we propose also represent belief rather than absolute truth.

5.1. Is the sensor broken or not?

First we suppose b_i represents the boolean property “sensor i is broken”. The formula $\max(b_i, \neg b_i)$ can be used to impose that the truth value of b_i is a binary value. An alternative could be to use the formula $b_i \oplus b_i \rightarrow b_i$. If b_i has truth value 1, it means that sensor i is broken. If it has truth value 0, the sensor works normally. Suppose that each sensor can only display a temperature in $[0, 1]$ with a limited granularity of one decimal but we have no idea how the grounding of decimal numbers is defined and that for a sensor that is not broken the maximum measurement error is 0.01. If for example the actual temperature is $t_i = 0.095$ and the sensor is not broken, then the measured temperature will be between 0.085 and 0.105. Since we do not know how the grounding works it is possible that the displayed temperature t'_i is equal to 0.2. In formulas (2)-(9) we will provide bounds on the actual temperatures that are large enough to take into account all possible scenarios. We can now write the following formulas with $i, j \in \{1, \dots, n\}$:

- (1) $\max(b_i, \neg b_i)$
- (2) $(0.2 \rightarrow_R \overline{t'_i}) \rightarrow (\text{B } b_i \oplus (0.09 \rightarrow_R t_i))$
- (3) $(0.3 \rightarrow_R \overline{t'_i}) \rightarrow (\text{B } b_i \oplus (0.19 \rightarrow_R t_i))$
- (4) \dots
- (5) $(1 \rightarrow_R \overline{t'_i}) \rightarrow (\text{B } b_i \oplus (0.89 \rightarrow_R t_i))$
- (6) $(\overline{t'_i} \rightarrow_R 0) \rightarrow (\text{B } b_i \oplus (t_i \rightarrow_R 0.11))$

- (7) $(\overline{t'_i} \rightarrow_R 0.1) \rightarrow (\text{B } b_i \oplus (t_i \rightarrow_R 0.21))$
- (8) \dots
- (9) $(\overline{t'_i} \rightarrow_R 0.8) \rightarrow (\text{B } b_i \oplus (t_i \rightarrow_R 0.91))$
- (10) $(\overline{w_{ij}} \rightarrow_R \neg(\overline{t'_i} \leftrightarrow \overline{t'_j})) \rightarrow (b_i \oplus b_j)$
- (11) $\neg \text{B } b_i \rightarrow \neg b_i$

Formulas (2) to (9) define the relationship between the measured and the actual temperature based on what is believed about sensors being broken or not and taking into account the granularity and the maximum measurement error. For instance, suppose you believe that sensor i is not broken and it is given that $t'_i = 0.4$. This means that there is an interpretation (I, S) such that $[\text{B } b_i]_{(I, S)} = 0$ and $I(\overline{t'_i}) = 0.4$. Formulas (2) to (5) then impose that $0.29 \leq I(t_i)$ and formulas (6) to (9) that $I(t_i) \leq 0.51$. If you believe that sensor i is broken, you cannot conclude anything about t_i .

Formula (10) imposes that if the difference between t'_i and t'_j is too large with respect to the weight w_{ij} , then at least one of the sensors must be broken. Formula (11) captures the connection between broken sensors and what you believe about them. It is needed to ensure that a sufficient number of sensors is believed to be broken, which in turn ensures that we do not derive more about the actual temperatures t_i than is warranted. In other words, (11) enforces some form of minimality.

Consider as a concrete example a forest with three sensors. Suppose we have $t'_1 = 0.4$, $t'_2 = 0.9$ and $t'_3 = 0.5$ and $w_{1,2} = 0.2$, $w_{1,3} = 0.2$ and $w_{2,3} = 0.2$.

We obtain the following degrees of similarity. For an arbitrary interpretation (I, S) we have

- $[\overline{t'_1} \leftrightarrow \overline{t'_2}]_{(I, S)} = 0.5$
- $[\overline{t'_1} \leftrightarrow \overline{t'_3}]_{(I, S)} = 0.9$
- $[\overline{t'_2} \leftrightarrow \overline{t'_3}]_{(I, S)} = 0.6$

For a structure (I, S) to model formulas (10) and (11) for each sensor i , it must hold that

- (a) $1 \leq I(b_1) + I(b_2)$
- (b) $1 \leq I(b_2) + I(b_3)$
- (c) $I(b_1) \leq [\text{B } b_1]_{(I, S)}$
- (d) $I(b_2) \leq [\text{B } b_2]_{(I, S)}$

$$(e) \ I(b_3) \leq [B b_3]_{(I,S)}$$

For a fuzzy autoepistemic model S of formulas (10) and (11) it holds by inequalities (c)-(e) that there is a unique fuzzy interpretation $I \in S$. Moreover it is a minimal fuzzy interpretation such that inequalities (a) and (b) are satisfied. It follows that there are only 2 fuzzy autoepistemic models of formulas (10) and (11):

$$S_1 = \{I \mid I(b_1) = 0, I(b_2) = 1 \text{ and } I(b_3) = 0\}$$

and

$$S_2 = \{I \mid I(b_1) = 1, I(b_2) = 0 \text{ and } I(b_3) = 1\}.$$

Finally, by including the formulas (2)-(9), we get two fuzzy autoepistemic models S_1 and S_2 :

$$S_1 = \{I \mid I(b_1) = 0, I(b_2) = 1, I(b_3) = 0,$$

$$0.29 \leq I(t_1) \leq 0.51, 0 \leq I(t_2) \leq 1 \text{ and } 0.39 \leq I(t_3) \leq 0.61\}$$

and

$$S_2 = \{I \mid I(b_1) = 1, I(b_2) = 0, I(b_3) = 1,$$

$$0 \leq I(t_1) \leq 1, 0.79 \leq I(t_2) \leq 1 \text{ and } 0 \leq I(t_3) \leq 1\}.$$

5.2. How big is the error on the measurement?

Now we suppose that e_i represents the error on the temperature measured by sensor i . We can then write the following formulas with $i, j \in \{1, \dots, n\}$:

1. $\overline{t'_i} \rightarrow (B e_i \oplus t_i)$
2. $\neg \overline{t'_i} \rightarrow (B e_i \oplus \neg t_i)$
3. $\neg(\overline{t'_i} \leftrightarrow \overline{t'_j}) \otimes \neg \overline{w_{ij}} \rightarrow (e_i \oplus e_j)$
4. $\neg B e_i \rightarrow \neg e_i$

Formulas (1) and (2) define the relationship between the measured and the actual temperature. For an interpretation (I, S) we have that $I(\overline{t'_i}) - [B e_i]_{(I,S)} \leq I(t_i) \leq I(\overline{t'_i}) + [B e_i]_{(I,S)}$.

Formula (3) imposes that if the difference between t'_i and t'_j is too big with respect to the weight w_{ij} , then there must be something wrong with the sensors. The size of the error depends on how big the difference between t'_i and t'_j is. Note that for an interpretation (I, S) we have $[\neg(\overline{t'_i} \leftrightarrow \overline{t'_j}) \otimes \neg \overline{w_{ij}}]_{(I,S)} = \max([\neg(\overline{t'_i} \leftrightarrow \overline{t'_j})]_I - [\overline{w_{ij}}]_I, 0)$.

Reconsider the previous example: $t'_1 = 0.4$, $t'_2 = 0.9$ and $t'_3 = 0.5$ and $w_{1,2} = 0.2$, $w_{1,3} = 0.2$ and $w_{2,3} = 0.2$.

For a structure (I, S) to model formulas (3) and (4) for each sensor i , it must hold that

$$(a) \quad 0.3 \leq I(e_1) + I(e_2)$$

$$(b) \quad 0.2 \leq I(e_2) + I(e_3)$$

$$(c) \quad I(e_1) \leq [B e_1]_{(I,S)}$$

$$(d) \quad I(e_2) \leq [B e_2]_{(I,S)}$$

$$(e) \quad I(e_3) \leq [B e_3]_{(I,S)}$$

For a fuzzy autoepistemic model S of formulas (3) and (4) it holds by inequalities (c)-(e) that there is a unique fuzzy interpretation $I \in S$. Moreover it is a minimal fuzzy interpretation such that inequalities (a) and (b) are satisfied. There are infinitely many fuzzy autoepistemic models S . However, we know by the minimality of $I \in S$ that $I(e_1)$ must be less or equal than 0.3, and $I(e_3)$ less or equal than 0.2.

Let us consider some examples.

$$S_1 = \{I \mid I(e_1) = 0.01, I(e_2) = 0.29, I(e_3) = 0,$$

$$0.39 \leq I(t_1) \leq 0.41, 0.6 \leq I(t_2) \leq 1 \text{ and } I(t_3) = 0.5\}$$

and

$$S_2 = \{I \mid I(e_1) = 0.29, I(e_2) = 0.01, I(e_3) = 0.19,$$

$$0.11 \leq I(t_1) \leq 0.69, 0.89 \leq I(t_2) \leq 0.91 \text{ and } 0.31 \leq I(t_3) \leq 0.69\}.$$

6. Relation between normal FASP and fuzzy autoepistemic logic

We will now show that the important relation between autoepistemic logic and ASP (Theorem 1) is preserved: fuzzy autoepistemic logic generalizes normal FASP. Suppose we have a normal program P in which each rule is of the form

$$r : a \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k),$$

with a an atom, b_i, c_j atoms and/or constants ($i \in \{1, \dots, m\}, j \in \{1, \dots, k\}$) and f an $(m + k)$ -ary function such that f is increasing in each of its arguments. The residual implicator associated with \leftarrow will be written as \mathbf{I} .

We use a similar transformation as for normal ASP. For rule r we define the associated fuzzy autoepistemic formula $\lambda(r)$ as

$$f(b_1, \dots, b_m, N_1(B c_1), \dots, N_k(B c_k)) \rightarrow a,$$

where N_j is the negation that is interpreted by the same negator \mathbb{N}_j as for not_j . The resulting fuzzy autoepistemic theory is $\lambda(P) = \{\lambda(r) \mid r \in P\}$. First we provide a lemma that characterizes the relationship between stable expansions of $\lambda(P)$ and stable expansions of the autoepistemic theory corresponding to a specific reduct of the program P .

Lemma 3. *Consider $E \in \mathcal{F}(\widetilde{L_B})$ and a normal FASP program P . Then E is a fuzzy stable expansion of $\lambda(P)$ iff E is a fuzzy stable expansion of $\lambda(P^{\hat{E}})$ with $\hat{E} = E|_{\mathcal{B}_P}$.*

Now we can prove the generalization of Theorem 1.

Theorem 2. *Consider a normal FASP program P . M is an answer set of P iff $\lambda(P)$ has a fuzzy stable expansion E such that $E|_{\mathcal{B}_P} = M$.*

Example 8. *Consider the FASP program P under the Lukasiewicz semantics from Example 5:*

$$\begin{aligned} b &\leftarrow \text{not } a \\ a &\leftarrow \text{not } b \end{aligned}$$

We compute the answer sets by using the characterization from Theorem 2. We compute the fuzzy stable expansions of $\lambda(P) = \{\neg(B a) \rightarrow b, \neg(B b) \rightarrow a\}$. Note that this is the fuzzy autoepistemic theory T we encountered in Examples 6 and 7. All fuzzy stable expansions of T are thus of the form E_x with $x \in [0, 1]$ such that $E_x(a) = x$, $E_x(b) = 1 - x$. Hence all answer sets are of the form M_x with $x \in [0, 1]$ such that $M_x(a) = x$ and $M_x(b) = 1 - x$. Note that for $x = 0$ and $x = 1$ we get the answer sets from Example 1.

Remark 6. *The result in Theorem 2 can be generalized to normal FASP programs also containing constraints i.e. rules with a constant in the head. If such rules may occur in a normal FASP program, it can be handled as follows. A fuzzy interpretation I is an answer set of a program $P \cup C$ with P a normal program with only atoms in the heads and C a set of constraints, iff I is an answer set of P and a fuzzy model of C . For example, consider the general FASP program from the previous example and add the constraint $0.3 \leftarrow a$. All answer sets are now of the form M_x with $x \in [0, 1]$ such that $M_x(a) = x$, $M_x(b) = 1 - x$ and $x \leq 0.3$.*

Remark 7. *Theorem 2 cannot be generalized to programs in which more complex formulas are allowed in heads of rules. For example, consider the general FASP program P containing the single rule*

$$a \oplus b \leftarrow \bar{1}$$

with a and b atoms and \oplus and \leftarrow connectives from Łukasiewicz logic. The corresponding fuzzy autoepistemic theory would be

$$\lambda(P) = \{\bar{1} \rightarrow a \oplus b\}.$$

All formulas in $\lambda(P)$ are objective, thus by Proposition 5, $\lambda(P)$ has only 1 fuzzy stable expansion. The program P however has infinitely many answer sets: for $x \in [0, 1]$, one can easily verify that $M_x(a) = x$ and $M_x(b) = 1 - x$ defines an answer set of P .

In Section 7 however, we will show that FASP can be embedded in fuzzy autoepistemic logic. We will return to this example in Example 9.

7. Relation between FASP and fuzzy autoepistemic logic

In this section we will investigate the relationship between FASP, the fuzzy logic of minimal belief and negation-as-failure (FMBNF) which we will define in Section 7.2 and the fuzzy autoepistemic logic introduced in Section 4. In particular, in Section 7.3 we will show how the answer sets of a general FASP program correspond to the models of a corresponding FMBNF theory. In Section 7.4, we will then use this result to embed FASP in fuzzy autoepistemic logic.

First we provide some background on the classical logic of minimal belief and negation-as-failure.

7.1. Logic of minimal belief and negation-as-failure (MBNF)

Lin and Shoham [9] defined a propositional nonmonotonic logic which uses two independent modal operators. One of them represents minimal belief and the other is related to the ideas of justification and of negation-as-failure. In this paper we consider a special case of such system: logic of minimal belief and negation-as-failure (MBNF)[10]. It extends the logic of grounded knowledge of Lin and Shoham [9] with the theories of epistemic queries by Levesque [52] and Reiter [53]. Formulas of the propositional logic of MBNF are built from a set of atoms A , the constants true and false, the

standard propositional connectives and two modal operators B and “not”. If a formula contains neither B nor “not” it is called *objective*. To make a clear distinction with autoepistemic logic, the formulas in MBNF are often called *bimodal*.

Truth is defined relative to a triple (I, S^b, S^n) with $I \in \mathcal{P}(A)$, $S^b \subseteq \mathcal{P}(A)$ and $S^n \subseteq \mathcal{P}(A)$. S^b is the set of possible worlds for defining the meaning of the operator B and S^n for “not”. If a formula is true in each world in S^b , it is believed by the agent. If there exists a world in S^n in which a formula is not true, then the agent does not believe it. In autoepistemic logic the sets S^n and S^b coincide. The interpretation of formulas is defined as usual, e.g.

- $(I, S^b, S^n) \models p$ iff $p \in I$
- $(I, S^b, S^n) \models (\neg\alpha)$ iff $(I, S^b, S^n) \not\models \alpha$
- $(I, S^b, S^n) \models B\alpha$ iff for every $J \in S^b$, $(J, S^b, S^n) \models \alpha$
- $(I, S^b, S^n) \models \text{not } \alpha$ iff for some $J \in S^n$, $(J, S^b, S^n) \not\models \alpha$

for $p \in A$ and a bimodal formula α .

Definition 7. Consider $I \in \mathcal{P}(A)$, $S \subseteq \mathcal{P}(A)$ and a MBNF theory T . The structure (I, S) is a MBNF model of T iff

1. for each $\alpha \in T$: $(I, S, S) \models \alpha$ and
2. there is no structure (I', S') such that $S \subset S'$ and $(I', S', S) \models \alpha$ for all $\alpha \in T$.

The maximality of S in Definition 7 expresses the idea of minimal belief: if the set of possible worlds is larger, then fewer propositions are believed. As Lin and Shoham pointed out, minimizing knowledge is not sufficient to model a rational agent’s knowledge or beliefs. Intuitively, the agent’s conclusions based on negation-as-failure should be supported by his knowledge.

Disjunctive ASP programs can be simulated by theories in MBNF. For each rule

$$r : a_1 \vee \dots \vee a_n \leftarrow b_1 \wedge \dots \wedge b_m \wedge \text{not } c_1 \wedge \dots \wedge \text{not } c_k,$$

in a disjunctive ASP program P , the formula $\mu(r)$

$$B b_1 \wedge \dots \wedge B b_m \wedge \text{not } c_1 \wedge \dots \wedge \text{not } c_k \rightarrow B a_1 \vee \dots \vee B a_n$$

is added to the theory $\mu(P)$ in MBNF. Lifschitz proved the following theorem with $\text{Mod}(M)$ the set of all interpretations I such that $M \subseteq I$.

Theorem 3. [10] Consider a disjunctive ASP program P . M is an answer set of P iff there exists a structure (I, S) which is a MBNF model of $\mu(P)$ such that $S = \text{Mod}(M)$.

Remark that in Theorem 3, the interpretation I is arbitrary, which follows easily from the fact that all occurrences of literals in $\mu(P)$ are encapsulated by either B or “not”. Using Theorem 3, Lifschitz and Schwarz [8], showed that disjunctive ASP programs (even with classical negation) can also be modelled in autoepistemic logic: For each rule

$$r : a_1 \vee \dots \vee a_n \leftarrow b_1 \wedge \dots \wedge b_m \wedge \text{not } c_1 \wedge \dots \wedge \text{not } c_k,$$

in a disjunctive ASP program P , the formula $\sigma(r)$

$$(b_1 \wedge B b_1) \wedge \dots \wedge (b_m \wedge B b_m) \wedge \text{not } c_1 \wedge \dots \wedge \text{not } c_k \rightarrow (a_1 \wedge B a_1) \vee \dots \vee (a_n \wedge B a_n)$$

is added to the autoepistemic theory $\sigma(P)$.

Theorem 4. [8] Consider a disjunctive ASP program P . M is an answer set of P iff $\text{Th}(\text{Mod}(M))$ is a stable expansion of $\sigma(P)$.

7.2. Fuzzy logic of minimal belief and negation-as-failure (FMBNF)

In a natural way we will generalize the definitions from Section 7.1 and use these to establish the relationship between FASP and fuzzy autoepistemic logic in Section 7.4. Formulas in fuzzy logic of minimal belief and negation-as-failure (FMBNF) are built from atoms in A , constants \bar{c} ($c \in [0, 1] \cap \mathbb{Q}$), n -ary connectives f ($n \in \mathbb{N}$) and the modal operators B and not_j . A *theory in FMBNF* is a set of formulas in FMBNF. If a formula contains neither B nor not_j it is called *objective*. Arbitrary formulas in FMBNF are called *bimodal*.

The semantics are defined relative to a triple (I, S^b, S^n) with $I \in \mathcal{F}(A)$, $S^b \subseteq \mathcal{F}(A)$ and $S^n \subseteq \mathcal{F}(A)$. S^b is the set of possible worlds for defining the meaning of B and S^n for the operators not_j . The interpretation of formulas is defined as follows:

- For atoms p : $[p]_{(I, S^b, S^n)} = I(p)$
- For constants \bar{c} : $[\bar{c}]_{(I, S^b, S^n)} = c$
- For a bimodal formula α : $[B \alpha]_{(I, S^b, S^n)} = \inf_{J \in S^b} [\alpha]_{(J, S^b, S^n)}$

- For a bimodal formula α : $[\text{not}_j \alpha]_{(I, S^b, S^n)} = \mathbb{N}_j \left(\inf_{J \in S^n} [\alpha]_{(J, S^b, S^n)} \right)$, where not_j is interpreted by the negator \mathbb{N}_j .
- For unimodal formulas α_i ($1 \leq i \leq n$) and $f \in F_n$ ($n \in \mathbb{N}$) which is interpreted by $\mathbf{f} : [0, 1]^n \rightarrow [0, 1]$:
 $[f(\alpha_1, \dots, \alpha_n)]_{(I, S^b, S^n)} = \mathbf{f}([\alpha_1]_{(I, S^b, S^n)}, \dots, [\alpha_n]_{(I, S^b, S^n)})$

Definition 8. Consider $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$. The structure (I, S) is a FMBNF model of a theory T in FMBNF iff

1. for each $\alpha \in T$: $[\alpha]_{(I, S, S)} = 1$ and
2. there is no structure (I', S') such that $S \subset S'$ and $[\alpha]_{(I', S', S)} = 1$ for all $\alpha \in T$.

For a fuzzy interpretation $I \in \mathcal{F}(A)$, we define $\text{Mod}(I)$ as the set of all fuzzy interpretations $K \in \mathcal{F}(A)$ such that $I \leq K$. With this definition, we have the following property.

Lemma 4. Consider $M, M' \in \mathcal{F}(A)$. Then $M < M'$ iff $\text{Mod}(M') \subset \text{Mod}(M)$.

Remark 8. For fuzzy sets $I \in \mathcal{F}(B)$ with $B \subset A$, we define $\text{Mod}(I) = \text{Mod}(\bar{I})$ with $\bar{I} \in \mathcal{F}(A)$ defined as follows: $\bar{I}(x) = I(x)$ if $x \in B$ and $\bar{I}(x) = 0$ if $x \notin B$. Lemma 4 then also holds for $M, M' \in \mathcal{F}(B)$.

7.3. Embedding FASP in FMBNF

First, we investigate the relationship between FMBNF and FASP. A general FASP program is translated to a theory in FMBNF as follows. Consider a general FASP program P . For each rule

$$r : g(a_1, \dots, a_n) \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k),$$

the formula $\mu(r)$

$$f(\mathbb{B} b_1, \dots, \mathbb{B} b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k) \rightarrow g(\mathbb{B} a_1, \dots, \mathbb{B} a_n)$$

is added to the theory $\mu(P)$ in FMBNF.

To prove the correspondence between the answer sets of P and the models of $\mu(P)$, we define for a general FASP program P and a fuzzy interpretation $M \in \mathcal{F}(\mathcal{B}_P)$, a set of fuzzy interpretations $\Pi(P, M) \subseteq \mathcal{F}(\mathcal{B}_P)$.

Definition 9. Consider a general FASP program P and $M \in \mathcal{F}(\mathcal{B}_P)$. Define

$$\pi_P^M = \{I \in \mathcal{F}(\mathcal{B}_P) \mid \forall \alpha \in \mu(P), \forall J \in \mathcal{F}(A) : [\alpha]_{(J, \text{Mod}(I), \text{Mod}(M))} = 1\}$$

and

$$\Pi(P, M) = \{I \in \mathcal{F}(\mathcal{B}_P) \mid I \text{ minimal such that } I \in \pi_P^M\}.$$

Note that π_P^M may not contain any minimal elements, and may even be empty. Accordingly, in such a situation, the program P will have no answer sets. Also, note that in Definition 9, since all atoms in $\mu(P)$ are preceded by a modal operator, the choice of the fuzzy interpretation J is irrelevant.

Lemma 5. Consider a positive FASP program P and $M \in \mathcal{B}_P$. Then $I \in \Pi(P, M)$ iff I is an answer set of P .

For a general FASP program we have the following result:

Lemma 6. Consider a general FASP program P and $M \in \mathcal{F}(\mathcal{B}_P)$. Then

$$\Pi(P^M, M) = \Pi(P, M).$$

Next, we will define a notion of equivalence for subsets of $\mathcal{F}(A)$.

Definition 10. Consider a FASP-program P and $S_1, S_2 \subseteq \mathcal{F}(A)$. We say that S_1 and S_2 are inf-equivalent w.r.t. P if $\forall x \in \mathcal{B}_P$

$$\inf_{I \in S_1} I(x) = \inf_{I \in S_2} I(x).$$

For inf-equivalent sets of fuzzy interpretations we have the following result.

Lemma 7. Consider a general FASP program P and $S_1, S_2 \subseteq \mathcal{F}(A)$ inf-equivalent w.r.t. P . Then for each $r \in P$ we have

$$[\mu(r)]_{(I, S_1, S)} = [\mu(r)]_{(I, S_2, S)},$$

with $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$ arbitrary.

We need one more lemma.

Lemma 8. Consider a general FASP program P , $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$. Then (I, S) is a FMBNF model of $\mu(P)$ iff there exists $M \in \Pi(P, M)$ such that $S = \text{Mod}(M)$.

Combining Lemmas 5-8, we get the following generalization of Theorem 3.

Theorem 5. Consider a general FASP program P , $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$. A structure (I, S) is a FMBNF model of $\mu(P)$ iff $S = \text{Mod}(M)$ with M an answer set of P .

7.4. Embedding FASP in fuzzy autoepistemic logic

Using the result from Theorem 5, we will now generalize Theorem 4: fuzzy autoepistemic logic generalizes FASP. The translation is done as follows. For each rule

$$r : g(a_1, \dots, a_n) \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k)$$

in a general FASP program P we add the formula $\sigma(r)$:

$$f(\min(b_1, B b_1), \dots, \min(b_m, B b_m), N_1(B c_1), \dots, N_k(B c_k)) \rightarrow \\ g(\min(a_1, B a_1), \dots, \min(a_n, B a_n)),$$

where N_j is the negation that is interpreted by the same negator N_j as for not_j to the fuzzy autoepistemic theory $\sigma(P)$.

First we provide some useful lemmas.

Lemma 9. *Consider a general FASP program P and $S \subseteq \mathcal{F}(A)$. Then for $r \in P$, $I, J \in \mathcal{F}(A)$ and $S' = S \cup \{J\}$ we have that*

$$[\mu(r)]_{(I, S', S)} = [\sigma(r)]_{(J, S)}.$$

Lemma 10. *Consider a general FASP program P , $I' \in \mathcal{F}(A)$ and $M, M' \in \mathcal{F}(A)$ such that $M' \leq M$. Then for $r \in P$ and for $S, S' \subseteq \mathcal{F}(A)$ such that S and $\text{Mod}(M)$ are inf-equivalent and S' and $\text{Mod}(M')$ are inf-equivalent w.r.t. P , we have that*

$$[\sigma(r)]_{(M', S)} = [\mu(r)]_{(I', S', S)}.$$

We will now use Lemmas 9 and 10 to prove the main theorem:

Theorem 6. *Consider a general FASP program P and $M \in \mathcal{F}(\mathcal{B}_P)$. M is an answer set of P iff $\text{Mod}(M)$ is a fuzzy autoepistemic model of $\sigma(P)$.*

Example 9. *Reconsider the general FASP program P from Remark 7.*

$$a \oplus b \leftarrow \bar{1}$$

with a and b atoms and \oplus and \leftarrow connectives from Lukasiewicz logic. The corresponding fuzzy autoepistemic theory is

$$\sigma(P) = \{\bar{1} \rightarrow \min(a, B a) \oplus \min(b, B b)\}.$$

We will use Theorem 6 to calculate all answer sets of P . Suppose $I \in \mathcal{F}(A)$ is an arbitrary answer set of P . Rename $I(a) = x$ and $I(b) = y$. By Theorem 6, it then follows that

$$S = \text{Mod}(I) = \{J \in \mathcal{F}(A) \mid J(a) \geq x, J(b) \geq y\}$$

is a fuzzy autoepistemic model of $\sigma(P)$. Thus, what we want to know is for which values x and y we have that S is a fuzzy autoepistemic model, or in other words

$$S = \left\{ J \in \mathcal{F}(A) \mid 1 \leq \min(J(a), \inf_{K \in S} K(a)) + \min(J(b), \inf_{K \in S} K(b)) \right\}.$$

For an arbitrary $J \in S$ we have

$$\begin{aligned} \min(J(a), \inf_{K \in S} K(a)) + \min(J(b), \inf_{K \in S} K(b)) &= \inf_{K \in S} K(a) + \inf_{K \in S} K(b) \\ &= x + y \end{aligned}$$

Hence $x + y \geq 1$ is a necessary condition. Now suppose that $x + y > 1$. We show that

$$\left\{ J \in \mathcal{F}(A) \mid 1 \leq \min(J(a), \inf_{K \in S} K(a)) + \min(J(b), \inf_{K \in S} K(b)) \right\} \not\subseteq S.$$

Indeed, consider $J \in \mathcal{F}(A)$ defined by $J(a) = x$ and $J(b) = 1 - x$. If $x + y > 1$ then

$$\begin{aligned} \min(J(a), \inf_{K \in S} K(a)) + \min(J(b), \inf_{K \in S} K(b)) &= \min(x, x) + \min(1 - x, y) \\ &= x + (1 - x) \\ &= 1 \end{aligned}$$

Thus J is an element of the set on the left hand side. If $J \in S$, then $1 - x = J(b) \geq y$ or $x + y \leq 1$, a contradiction. Hence $J \notin S$.

8. Conclusions

In this paper we have introduced a fuzzy version of autoepistemic logic, which can be used to reason about one's beliefs about the degrees to which properties are satisfied. The general theory of fuzzy autoepistemic logic is

also useful for abductive reasoning about theories with gradual propositions. We have shown that important properties of classical autoepistemic logic are preserved and that the relation between answer set programming and autoepistemic logic remains valid when generalizing to fuzzy logics. Moreover, we have presented two different but equivalent characterizations of answer sets in fuzzy autoepistemic logic and in fuzzy logic of minimal belief and negation-as-failure. These results lead to a better comprehension of how to interpret fuzzy answer sets.

In future work, it would be interesting to see whether the implementation of classical autoepistemic logic by using quantified boolean formulas [54] can be extended to fuzzy logics using multi-level linear programming. If this is indeed the case, it could be used as a basis to implement fuzzy autoepistemic logic reasoners, as well as fuzzy answer set programming solvers. In practice, for a general FASP program P we are interested in one of following problems: finding an arbitrary answer set of P , finding an answer set I of P such that for a particular atom a and a value $\lambda \in [0, 1] \cap \mathbb{Q}$ it holds that $I(a) \geq \lambda$ and checking if for a particular atom a and a value $\lambda \in [0, 1] \cap \mathbb{Q}$ it holds that $I(a) \geq \lambda$ for each answer set I of P . Although it is not always possible to list all answer sets of a program since there may be infinitely many, in [55] we show that for an important type of general FASP programs we can solve these problems using bilevel programming.

Appendix A. Proofs

Proof of Lemma 1

Lemma. *Let $Q \subseteq \mathcal{F}(A')$ such that for each $I' \in Q$ and $\varphi \in \widetilde{L}_B$ we have that $[B\varphi]_{I'} = \inf_{J' \in Q} [\varphi]_{J'}$. Then, for $\alpha \in \widetilde{L}_B$, $S = \{I'_{|A} \mid I' \in Q\}$ and $I' \in Q$ we have*

$$[\alpha]_{(I'_{|A}, S)} = [\alpha]_{I'}. \quad (\text{A.1})$$

Proof. If α is an objective formula, (A.1) is satisfied by the definition of the satisfaction relation. For unimodal formulas, we will prove this by structural induction.

Consider $B\varphi$ such that (A.1) is satisfied for φ . For each $I' \in Q$ it holds by definition of Q that $[B\varphi]_{I'} = \inf_{J' \in Q} [\varphi]_{J'}$. By the induction hypothesis, the latter is equal to $\inf_{J' \in Q} [\varphi]_{(J'_{|A}, S)}$. By definition of S we have

$\inf_{J' \in Q} [\varphi]_{(J'_A, S)} = \inf_{K \in S} [\varphi]_{(K, S)}$. Finally by definition of the interpretation of unimodal formulas, we have $\inf_{K \in S} [\varphi]_{(K, S)} = [\mathbf{B} \varphi]_{(I'_A, S)}$.

Now suppose $\alpha = f(\varphi_1, \dots, \varphi_n)$ such that (A.1) is satisfied for φ_i ($i \in \{1, \dots, n\}$). We conclude that

$$\begin{aligned} [f(\varphi_1, \dots, \varphi_n)]_{(I'_A, S)} &= \mathbf{f}([\varphi_1]_{(I'_A, S)}, \dots, [\varphi_n]_{(I'_A, S)}) \\ &= \mathbf{f}([\varphi_1]_{I'}, \dots, [\varphi_n]_{I'}) = [f(\varphi_1, \dots, \varphi_n)]_{I'}. \end{aligned}$$

□

Proof of Lemma 2

Lemma. *Let $S \subseteq \mathcal{F}(A)$ and $I' \in \mathcal{F}(A')$ such that for each $\varphi \in \widetilde{L}_B$ we have $[\mathbf{B} \varphi]_{I'} = \text{Th}(S)(\varphi)$. Then, for $\alpha \in \widetilde{L}_B$ we have*

$$[\alpha]_{(I'_A, S)} = [\alpha]_{I'}. \quad (\text{A.2})$$

Proof. For objective formulas, (A.2) is satisfied by the definition of the satisfaction relation. For unimodal formulas, we will prove this by structural induction.

For formulas $\mathbf{B} \varphi$ with $\varphi \in \widetilde{L}_B$ it follows easily. Indeed, by definition of S and by Definition 6, we have $[\mathbf{B} \varphi]_{I'} = \text{Th}(S)(\varphi) = \inf_{K \in S} [\varphi]_{(K, S)}$. By definition of the interpretation of unimodal formulas, the latter is equal to $[\mathbf{B} \varphi]_{(I'_A, S)}$. Thus we have shown

$$[\mathbf{B} \varphi]_{(I'_A, S)} = [\mathbf{B} \varphi]_{I'}.$$

Formulas of the form $f(\alpha_1, \dots, \alpha_n)$ are handled as in the proof of Lemma 1. □

Proof of Proposition 4

Proposition. *Consider a fuzzy autoepistemic theory T in \widetilde{L}_B .*

1. *If $E \in \mathcal{F}(\widetilde{L}_B)$ is a fuzzy stable expansion of T , then $E = \text{Th}(S)$ with S a fuzzy autoepistemic model of T .*
2. *If $S \subseteq \mathcal{F}(A)$ is a fuzzy autoepistemic model of T , then $E = \text{Th}(S)$ is a fuzzy stable expansion of T .*

Proof. We prove this statement in two parts.

1. Suppose E is a fuzzy stable expansion of T . By Definition 4 we have for $\alpha \in \widetilde{L}_B$

$$[B \alpha]_{J'} = E(\alpha) = \inf_{I' \in Q} [\alpha]_{I'},$$

with

$$Q = \left\{ I' \in \mathcal{F}(A') \mid I' \models T \text{ and } [B \varphi]_{I'} = E(\varphi) \text{ for all } \varphi \in \widetilde{L}_B \right\}$$

and $J' \in Q$ arbitrary. Let us define

$$S = \{ I'_A \mid I' \in Q \}.$$

We prove that S is a fuzzy autoepistemic model of T and that $E = \text{Th}(S)$. By Lemma 1 we have that for $\alpha \in \widetilde{L}_B$ and $I' \in Q$

$$[\alpha]_{(I'_A, S)} = [\alpha]_{I'}.$$

We can now prove that S is a fuzzy autoepistemic model of T . For $I \in S$ we have that $I = I'_A$ for some $I' \in Q$. For every $\varphi \in T$, it then holds that

$$[\varphi]_{(I, S)} = [\varphi]_{(I'_A, S)} = [\varphi]_{I'} = 1,$$

where the last equality follows from the observation that $I' \models T$. We have now shown

$$S \subseteq \{ I \in \mathcal{F}(A) \mid \forall \varphi \in T : [\varphi]_{(I, S)} = 1 \}.$$

For the converse inclusion, consider an interpretation $I \in \mathcal{F}(A)$ such that $[\varphi]_{(I, S)} = 1$ for each $\varphi \in T$. We will now construct $I' \in Q$ such that $I'_A = I$. For $a \in A$, we define $I'(a) = I(a)$ and for a formula $\alpha \in \widetilde{L}_B$, we define $[B \alpha]_{I'} = E(\alpha)$. It is then clear that $I'_A = I$. Indeed, by definition of I and by Lemma 1 it follows that for each $\alpha \in T$ we have $[\alpha]_{I'} = [\alpha]_{(I'_A, S)} = [\alpha]_{(I, S)} = 1$. Hence $I' \models T$ and thus $I' \in Q$. It follows that

$$S = \{ I \in \mathcal{F}(A) \mid \forall \varphi \in T : [\varphi]_{(I, S)} = 1 \}.$$

Finally, we prove that $E(\alpha) = \text{Th}(S)(\alpha)$ for each unimodal formula α . For any $I' \in Q$ we have by definition that $E(\alpha) = [B \alpha]_{I'}$ and by Lemma

1 that $[B\alpha]_{I'} = [B\alpha]_{(I'_A, S)}$. We also have $[B\alpha]_{(I'_A, S)} = \inf_{K \in S} [\alpha]_{(K, S)}$, which is equal to $\text{Th}(S)(\alpha)$. We conclude

$$E(\alpha) = \text{Th}(S)(\alpha).$$

2. Now suppose we have a fuzzy autoepistemic model S of T . For each unimodal formula α , we define $E(\alpha)$ as follows

$$E(\alpha) = \inf_{I' \in M} [\alpha]_{I'},$$

with

$$M = \left\{ I' \in \mathcal{F}(A') \mid I'_A \in S \text{ and } \forall \varphi \in \widetilde{L}_B : [B\varphi]_{I'} = \text{Th}(S)(\varphi) \right\}.$$

We show that $E = \text{Th}(S)$ and that E is a fuzzy stable expansion of T . By Lemma 2, it follows that for $I' \in M$ and $\alpha \in \widetilde{L}_B$

$$[\alpha]_{(I'_A, S)} = [\alpha]_{I'}.$$

Note that, similar as earlier in the proof, for each $I \in S$ we can find $I' \in M$ such that $I'_A = I$. Thus by definition of M we have $S = \{I'_A \mid I' \in M\}$. These two observations lead to the equality $E = \text{Th}(S)$. Indeed, for $\alpha \in \widetilde{L}_B$, we have

$$\text{Th}(S)(\alpha) = \inf_{I \in S} [\alpha]_{(I, S)} = \inf_{I' \in M} [\alpha]_{(I'_A, S)} = \inf_{I' \in M} [\alpha]_{I'} = E(\alpha).$$

To see that E is a fuzzy stable expansion of T , it is sufficient to prove that M is equal to

$$P = \left\{ I' \in \mathcal{F}(A') \mid I' \models T \text{ and } [B\varphi]_{I'} = E(\varphi) \text{ for all } \varphi \in \widetilde{L}_B \right\}.$$

First, we show $M \subseteq P$. Let $I' \in M$. By Lemma 2, it follows that $I' \models T$: for $\alpha \in T$ we have $[\alpha]_{I'} = [\alpha]_{(I'_A, S)} = 1$, where the last equality holds since S is a fuzzy autoepistemic model of T . It remains to be shown that $[B\varphi]_{I'} = E(\varphi)$ for each $\varphi \in \widetilde{L}_B$. This follows easily:

$$E(\varphi) = \text{Th}(S)(\varphi) = \inf_{I \in S} [\varphi]_{(I, S)} = [B\varphi]_{(I'_A, S)} = [B\varphi]_{I'},$$

where the last equality follows from Lemma 2.

To conclude the proof, we show that $P \subseteq M$. Recall that we already showed $E = \text{Th}(S)$. It follows that

$$P \subseteq \left\{ I' \in \mathcal{F}(A') \mid \forall \varphi \in \widetilde{L}_B : [\text{B } \varphi]_{I'} = \text{Th}(S)(\varphi) \right\}.$$

To show that $P \subseteq M$, it is sufficient to show that for $I' \in P$ it holds that $I'_{|A} \in S$. From Lemma 2 and the fact that $I' \models T$, we have that

$$[\alpha]_{(I'_{|A}, S)} = [\alpha]_{I'} = 1,$$

for $\alpha \in T$. This means that $I'_{|A} \in S$ since S is a fuzzy autoepistemic model of T .

□

Proof of Proposition 5

Proposition. *Every set of objective formulas in \widetilde{L}_B has a unique fuzzy stable expansion.*

Proof. By Definition 5, S is a fuzzy autoepistemic model S of T iff

$$S = \left\{ I \in \mathcal{F}(A) \mid \forall \varphi \in T : [\varphi]_{(I, S)} = 1 \right\}.$$

Since T only contains objective formulas, we have for $\varphi \in T$ that $[\varphi]_{(I, S)} = [\varphi]_I$. Thus

$$S = \left\{ I \in \mathcal{F}(A) \mid \forall \varphi \in T : [\varphi]_I = 1 \right\}$$

is the unique fuzzy autoepistemic model of T . By Proposition 4 there is exactly one fuzzy stable expansion.

□

Proof of Lemma 3

Lemma. *Consider $E \in \mathcal{F}(\widetilde{L}_B)$ and a normal FASP program P . Then E is a fuzzy stable expansion of $\lambda(P)$ iff E is a fuzzy stable expansion of $\lambda(P^{\hat{E}})$ with $\hat{E} = E_{|B_P}$.*

Proof. We need to prove that for each $\alpha \in \widetilde{L}_B$

$$\begin{aligned} & \inf \left\{ [\alpha]_{I'} \mid I' \models \lambda(P) \text{ and } [B\varphi]_{I'} = E(\varphi) \text{ for all } \varphi \in \widetilde{L}_B, I' \in \mathcal{F}(A') \right\} \\ &= \inf \left\{ [\alpha]_{I'} \mid I' \models \lambda(P^{\hat{E}}) \text{ and } [B\varphi]_{I'} = E(\varphi) \text{ for all } \varphi \in \widetilde{L}_B, I' \in \mathcal{F}(A') \right\} \end{aligned}$$

We will show this by proving that $I' \models \lambda(P)$ iff $I' \models \lambda(P^{\hat{E}})$ as soon as $[B\varphi]_{I'} = E(\varphi)$ for each $\varphi \in \widetilde{L}_B$.

Indeed, for such interpretations I' we have

$$[N_j(Bc)]_{I'} = N_j([Bc]_{I'}) = N_j(E(c)) = N_j(\hat{E}(c)) = [\text{not}_j c]_{\hat{E}},$$

for $c \in \mathcal{B}_P$. Thus, for a rule r

$$a \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k)$$

we have

$$\begin{aligned} [\lambda(r^{\hat{E}})]_{I'} &= [f(b_1, \dots, b_m, \overline{[\text{not}_1 c_1]_{\hat{E}}}, \dots, \overline{[\text{not}_k c_k]_{\hat{E}}}) \rightarrow a]_{I'} \\ &= I\left(\mathbf{f}(I'(b_1), \dots, I'(b_m), [\text{not}_1 c_1]_{\hat{E}}, \dots, [\text{not}_k c_k]_{\hat{E}}), I'(a)\right) \\ &= I\left(\mathbf{f}(I'(b_1), \dots, I'(b_m), [N_1(Bc_1)]_{I'}, \dots, [N_k(Bc_k)]_{I'}), I'(a)\right) \\ &= [f(b_1, \dots, b_m, N_1(Bc_1), \dots, N_k(Bc_k)) \rightarrow a]_{I'} \\ &= [\lambda(r)]_{I'} \end{aligned}$$

□

Proof of Theorem 2

Theorem. Consider a normal FASP program P . M is an answer set of P iff $\lambda(P)$ has a fuzzy stable expansion E such that $E|_{\mathcal{B}_P} = M$.

Proof. (\Rightarrow) First suppose M is an answer set of P . By definition, M is the minimal fuzzy model of P^M and thus also the minimal fuzzy model of $\lambda(P^M)$ since P^M is a program without negation-as-failure. Since $\lambda(P^M)$ is a set of objective formulas, it has exactly one fuzzy stable expansion E (Proposition 5). By Proposition 4 we know that $E = \text{Th}(S)$ with S the fuzzy autoepistemic model of $\lambda(P^M)$:

$$S = \{I \in \mathcal{F}(A) \mid \forall \alpha \in \lambda(P^M) : [\alpha]_I = 1\}.$$

Since S contains all fuzzy models of $\lambda(P^M)$ and M is the unique minimal fuzzy model we have

$$E(a) = \text{Th}(S)(a) = \inf_{I \in S} I(a) = M(a)$$

for $a \in \mathcal{B}_P$. Hence $E|_{\mathcal{B}_P} = M$. It follows that E is a fuzzy stable expansion of $\lambda(P^M) = \lambda(P^{\hat{E}})$, with $\hat{E} = E|_{\mathcal{B}_P}$. By Lemma 3 we conclude that E is a fuzzy stable expansion of $\lambda(P)$.

(\Leftarrow) Now suppose that $\lambda(P)$ has a fuzzy stable expansion E . We prove that $M = E|_{\mathcal{B}_P} = \hat{E}$ is an answer set of P by showing that M is the minimal fuzzy model of $P^{\hat{E}}$ or thus the minimal fuzzy model of $\lambda(P^{\hat{E}})$ since $P^{\hat{E}}$ is a program without negation-as-failure. By Lemma 3 we know that E is a fuzzy stable expansion of $\lambda(P^{\hat{E}})$. This implies by Proposition 4 that $E = \text{Th}(S)$ with

$$S = \left\{ I \in \mathcal{F}(A) \mid \forall \alpha \in \lambda(P^{\hat{E}}) : [\alpha]_I = 1 \right\}.$$

We now prove that M is a fuzzy model of $\lambda(P^{\hat{E}})$ by using the definition of S : for $\alpha \in \lambda(P^{\hat{E}})$ we have

$$[\alpha]_M = [\alpha]_{\hat{E}} = E(\alpha) = \text{Th}(S)(\alpha) = \inf_{I \in S} [\alpha]_{(I,S)} = 1.$$

To show that M is the minimal fuzzy model, we consider an arbitrary fuzzy model I of $\lambda(P^{\hat{E}})$. Since S contains all fuzzy models of $\lambda(P^{\hat{E}})$, I must be an element of S . For $a \in \mathcal{B}_P$ we then have

$$M(a) = \hat{E}(a) = E(a) = \text{Th}(S)(a) = \inf_{K \in S} K(a) \leq I(a).$$

□

Proof of Lemma 4

Lemma. Consider $M, M' \in \mathcal{F}(A)$. Then $M < M'$ iff $\text{Mod}(M') \subset \text{Mod}(M)$.

Proof. Remark that $M \leq M'$ iff $\text{Mod}(M') \subseteq \text{Mod}(M)$. Suppose that $M < M'$. Since $M \leq M'$ it follows that $\text{Mod}(M') \subseteq \text{Mod}(M)$. If $\text{Mod}(M') = \text{Mod}(M)$, then it would follow that $M' \leq M$, a contradiction. Similary, $\text{Mod}(M') \subset \text{Mod}(M)$ implies that $M < M'$. □

Proof of Lemma 5

Lemma. Consider a positive FASP program P and $M \in \mathcal{B}_P$. Then $I \in \Pi(P, M)$ iff I is an answer set of P .

Proof. We need to prove that $\Pi(P, M)$ is exactly the set of the minimal elements of $\{I \in \mathcal{F}(\mathcal{B}_P) \mid I \models P\}$. By definition of $\Pi(P, M)$, it is thus sufficient to prove that $I \in \pi_P^M$ iff $I \models P$.

A rule r in P is of the form

$$r : g(a_1, \dots, a_n) \leftarrow f(b_1, \dots, b_m).$$

Note that for $I \in \mathcal{F}(\mathcal{B}_P)$ we have $I(a) = \inf_{K \in \text{Mod}(I)} K(a)$ for each $a \in \mathcal{B}_P$. The statement then follows:

$$\begin{aligned} & I \models P \\ \Leftrightarrow & \forall r \in P : \mathbf{f}(I(b_1), \dots, I(b_m)) \leq \mathbf{g}(I(a_1), \dots, I(a_n)) \\ \Leftrightarrow & \forall r \in P : \mathbf{f}\left(\inf_{K \in \text{Mod}(I)} K(b_1), \dots, \inf_{K \in \text{Mod}(I)} K(b_m)\right) \\ & \leq \mathbf{g}\left(\inf_{K \in \text{Mod}(I)} K(a_1), \dots, \inf_{K \in \text{Mod}(I)} K(a_n)\right) \\ \Leftrightarrow & \forall r \in P, \forall J \in \mathcal{F}(A) : \\ & \mathbf{f}([\mathbf{B} b_1]_{(J, \text{Mod}(I), \text{Mod}(M))}, \dots, [\mathbf{B} b_m]_{(J, \text{Mod}(I), \text{Mod}(M))}) \\ & \leq \mathbf{g}([\mathbf{B} a_1]_{(J, \text{Mod}(I), \text{Mod}(M))}, \dots, [\mathbf{B} a_n]_{(J, \text{Mod}(I), \text{Mod}(M))}) \\ \Leftrightarrow & \forall r \in P, \forall J \in \mathcal{F}(A) : \\ & [f(\mathbf{B} b_1, \dots, \mathbf{B} b_m) \rightarrow g(\mathbf{B} a_1, \dots, \mathbf{B} a_n)]_{(J, \text{Mod}(I), \text{Mod}(M))} = 1 \\ \Leftrightarrow & \forall \alpha \in \mu(P), \forall J \in \mathcal{F}(A) : \\ & [\alpha]_{(J, \text{Mod}(I), \text{Mod}(M))} = 1 \end{aligned}$$

□

Proof of Lemma 6

Lemma. Consider a general FASP program P and $M \in \mathcal{F}(\mathcal{B}_P)$. Then

$$\Pi(P^M, M) = \Pi(P, M).$$

Proof. By Definition 9, it is sufficient to check that for $I \in \mathcal{B}_P$:

$$\forall \alpha \in \mu(P), \forall J \in \mathcal{F}(A) : [\alpha]_{(J, \text{Mod}(I), \text{Mod}(M))} = 1$$

is equivalent with

$$\forall \alpha \in \mu(P^M), \forall J \in \mathcal{F}(A) : [\alpha]_{(J, \text{Mod}(I), \text{Mod}(M))} = 1$$

Hence for a rule in P

$$r : g(a_1, \dots, a_n) \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k)$$

it is sufficient to prove that for each $J \in \mathcal{F}(A)$ we have

$$[\mu(r)]_{(J, \text{Mod}(I), \text{Mod}(M))} = 1 \Leftrightarrow [\mu(r^M)]_{(J, \text{Mod}(I), \text{Mod}(M))} = 1.$$

Thus we need to show that for each atom c_j ($j \in \{1, \dots, k\}$) we have

$$[\text{not}_j c_j]_{(J, \text{Mod}(I), \text{Mod}(M))} = [\text{B}(\overline{[\text{not}_j c_j]_M})]_{(J, \text{Mod}(I), \text{Mod}(M))}.$$

But this follows easily by the definitions and since $\overline{[\text{not}_j c_j]_M}$ is a constant:

$$\begin{aligned} [\text{B}(\overline{[\text{not}_j c_j]_M})]_{(J, \text{Mod}(I), \text{Mod}(M))} &= \inf_{K \in \text{Mod}(I)} [\overline{[\text{not}_j c_j]_M}]_{(K, \text{Mod}(I), \text{Mod}(M))} \\ &= [\text{not}_j c_j]_M \\ &= \mathbb{N}_j(M(c_j)) \\ &= \mathbb{N}_j \left(\inf_{K \in \text{Mod}(M)} K(c_j) \right) \\ &= [\text{not}_j c_j]_{(J, \text{Mod}(I), \text{Mod}(M))} \end{aligned}$$

□

Proof of Lemma 7

Lemma. Consider a general FASP program P and $S_1, S_2 \subseteq \mathcal{F}(A)$ inf-equivalent w.r.t. P . Then for each $r \in P$ we have

$$[\mu(r)]_{(I, S_1, S)} = [\mu(r)]_{(I, S_2, S)},$$

with $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$ arbitrary.

Proof. The lemma follows straightforwardly from the way in which the semantics of the modality B is defined in FMBNF. □

Proof of Lemma 8

Lemma. Consider a general FASP program P , $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$. Then (I, S) is a FMBNF model of $\mu(P)$ iff there exists $M \in \Pi(P, M)$ such that $S = \text{Mod}(M)$.

Proof. (\Rightarrow) First assume that (I, S) is a FMBNF model of $\mu(P)$. By Definition 8, this means that

- (a) for each $\alpha \in \mu(P)$: $[\alpha]_{(I, S, S)} = 1$ and
- (b) there is no structure (I', S') such that $S \subset S'$ and $[\alpha]_{(I', S', S)} = 1$ for all $\alpha \in \mu(P)$.

Define

$$M : \mathcal{B}_P \rightarrow [0, 1] : x \mapsto \inf_{J \in S} J(x).$$

For $K \in S$, we then have $M(x) = \inf_{J \in S} J(x) \leq K(x)$ for all $x \in \mathcal{B}_P$. Since $\bar{M}(x) = 0 \leq K(x)$ for $x \notin \mathcal{B}_P$, we conclude that $S \subseteq \text{Mod}(M)$. By definition of M , we have that $\inf_{J \in S} J(x) = M(x) = \inf_{J \in \text{Mod}(M)} J(x)$ for all $x \in \mathcal{B}_P$. Hence S and $\text{Mod}(M)$ are inf-equivalent w.r.t P . By Lemma 7, it follows that $[\alpha]_{(I, \text{Mod}(M), S)} = [\alpha]_{(I, S, S)} = 1$ for all $\alpha \in \mu(P)$. Now suppose that $S \subset \text{Mod}(M)$, then we have a contradiction since S is maximal under all S' such that $[\alpha]_{(I, S', S)} = 1$ for all $\alpha \in \mu(P)$. Thus $S = \text{Mod}(M)$ and $[\alpha]_{(I, \text{Mod}(M), \text{Mod}(M))} = 1$ for all $\alpha \in \mu(P)$. To prove that $M \in \Pi(P, M)$ it remains to show that M is minimal under all $M' \in \mathcal{F}(\mathcal{B}_P)$ such that for all $J \in F(A)$ and for all $\alpha \in \mu(P)$ it holds that $[\alpha]_{(J, \text{Mod}(M'), \text{Mod}(M))} = 1$. Suppose this is not the case and there exists an $M' \in \mathcal{F}(\mathcal{B}_P)$ such that $M' < M$ and for all $J \in F(\mathcal{B}_P)$ and for all $\alpha \in \mu(P)$ it holds that $[\alpha]_{(J, \text{Mod}(M'), \text{Mod}(M))} = 1$. By Lemma 4, it follows that $S = \text{Mod}(M) \subset \text{Mod}(M')$. This is in contradiction with the maximality of S .

(\Leftarrow) Now assume $S = \text{Mod}(M)$ with $M \in \Pi(P, M)$. By Definition 9, this means that M is minimal under all $M' \in \mathcal{F}(\mathcal{B}_P)$ such that for all $\alpha \in \mu(P)$ and for all $J \in \mathcal{F}(A)$ it holds that $[\alpha]_{(J, \text{Mod}(M'), \text{Mod}(M))} = 1$. Thus, we already know that $[\alpha]_{(I, S, S)} = [\alpha]_{(I, \text{Mod}(M), \text{Mod}(M))} = 1$ for all $\alpha \in \mu(P)$. To show that (I, S) is a FMBNF model of $\mu(P)$, it remains to be showed that there is no structure (I', S') such that $\text{Mod}(M) \subset S'$ and $[\alpha]_{(I', S', \text{Mod}(M))} = 1$ for all $\alpha \in \mu(P)$. Suppose there exists such a structure (I', S') . Define

$$M' : \mathcal{B}_P \rightarrow [0, 1] : x \mapsto \inf_{J \in S'} J(x).$$

Similar as in the first part of the proof we have that $S' \subseteq \text{Mod}(M')$ and that S' and $\text{Mod}(M')$ are inf-equivalent w.r.t. P . By Lemma 7 we then have that $[\alpha]_{(I', \text{Mod}(M'), \text{Mod}(M))} = [\alpha]_{(I', S', \text{Mod}(M))} = 1$ for all $\alpha \in \mu(P)$. Since $\mu(P)$ only contains atoms preceded by a modal operator, it also follows that for all $J \in \mathcal{F}(\mathcal{B}_P)$ and for all $\alpha \in \mu(P)$ we have $[\alpha]_{(J, \text{Mod}(M'), \text{Mod}(M))} = 1$. But this contradicts the minimality of M . Indeed, since $\text{Mod}(M) \subset S' \subseteq \text{Mod}(M')$, it follows by Lemma 4 that $M' < M$.

□

Proof of Theorem 5

Theorem 7. *Consider a general FASP program P , $I \in \mathcal{F}(A)$ and $S \subseteq \mathcal{F}(A)$. A structure (I, S) is a FMBNF model of $\mu(P)$ iff $S = \text{Mod}(M)$ with M an answer set of P .*

Proof.

$$\begin{aligned}
& M \text{ is an answer set of } P \\
& \Leftrightarrow M \text{ is an answer set of } P^M && \text{(definition answer set)} \\
& \Leftrightarrow M \in \Pi(P^M, M) && \text{(Lemma 5)} \\
& \Leftrightarrow M \in \Pi(P, M) && \text{(Lemma 6)} \\
& \Leftrightarrow (I, \text{Mod}(M)) \text{ is a FMBNF model of } \mu(P) && \text{(Lemma 8)}
\end{aligned}$$

□

Proof of Lemma 9

Lemma 11. *Consider a general FASP program P and $S \subseteq \mathcal{F}(A)$. Then for $r \in P$, $I, J \in \mathcal{F}(A)$ and $S' = S \cup \{J\}$ we have that*

$$[\mu(r)]_{(I, S', S)} = [\sigma(r)]_{(J, S)}.$$

Proof. Consider a rule $r \in P$

$$r : g(a_1, \dots, a_n) \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k).$$

Since $S' = S \cup \{J\}$ we have that $\inf_{K \in S'} K(a) = \min(J(a), \inf_{K \in S} K(a))$ for each $a \in A$. It follows that

$$\begin{aligned}
[\mu(r)]_{(I, S', S)} &= [f(\text{B } b_1, \dots, \text{B } b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k) \rightarrow g(\text{B } a_1, \dots, \text{B } a_n)]_{(I, S', S)} \\
&= \mathbf{I} \left(\mathbf{f} \left(\inf_{K \in S'} K(b_1), \dots, \inf_{K \in S'} K(b_m), N_1 \left(\inf_{K \in S} K(c_1) \right), \right. \right. \\
&\quad \left. \left. \dots, N_k \left(\inf_{K \in S} K(c_k) \right) \right), \mathbf{g} \left(\inf_{K \in S'} K(a_1), \dots, \inf_{K \in S'} K(a_n) \right) \right) \\
&= \mathbf{I} \left(\mathbf{f} \left(\min \left(J(b_1), \inf_{K \in S} K(b_1) \right), \dots, \min \left(J(b_m), \inf_{K \in S} K(b_m) \right), \right. \right. \\
&\quad \left. \left. N_1 \left(\inf_{K \in S} K(c_1) \right), \dots, N_k \left(\inf_{K \in S} K(c_k) \right) \right), \right. \\
&\quad \left. \mathbf{g} \left(\min \left(J(a_1), \inf_{K \in S} K(a_1) \right), \dots, \min \left(J(a_n), \inf_{K \in S} K(a_n) \right) \right) \right) \\
&= [f(\min(b_1, \text{B } b_1), \dots, \min(b_m, \text{B } b_m), N_1(\text{B } c_1), \dots, N_k(\text{B } c_k)) \\
&\quad \rightarrow g(\min(a, \text{B } a_1), \dots, \min(a_n, \text{B } a_n))]_{(J, S)} \\
&= [\sigma(r)]_{(J, S)}
\end{aligned}$$

□

Proof of Lemma 10

Lemma. Consider a general FASP program P , $I' \in \mathcal{F}(A)$ and $M, M' \in \mathcal{F}(A)$ such that $M' \leq M$. Then for $r \in P$ and for $S, S' \subseteq \mathcal{F}(A)$ such that S and $\text{Mod}(M)$ are inf-equivalent and S' and $\text{Mod}(M')$ are inf-equivalent w.r.t. P , we have that

$$[\sigma(r)]_{(M', S)} = [\mu(r)]_{(I', S', S)}.$$

Proof. Consider a rule $r \in P$:

$$r : g(a_1, \dots, a_n) \leftarrow f(b_1, \dots, b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k).$$

Since S and $\text{Mod}(M)$ are inf-equivalent w.r.t. P (Definition 10), we have $M(a) = \inf_{I \in \text{Mod}(M)} I(a) = \inf_{I \in S} I(a)$ for each $a \in \mathcal{B}_P$. By the inf-equivalence

of S' and $\text{Mod}(M')$, we have $M'(a) = \inf_{I \in \text{Mod}(M')} I(a) = \inf_{I \in S'} I(a)$ for each $a \in \mathcal{B}_P$. It follows that

$$\begin{aligned}
& [\sigma(r)]_{(M', S)} \\
&= [f(\min(b_1, B b_1), \dots, \min(b_m, B b_m), N_1(B c_1), \dots, N_k(B c_k)) \\
&\quad \rightarrow g(\min(a_1, B a_1), \dots, \min(a_n, B a_n))]_{(M', S)} \\
&= I \left(f \left(\min \left(M'(b_1), \inf_{K \in S} K(b_1) \right), \dots, \min \left(M'(b_m), \inf_{K \in S} K(b_m) \right), \right. \right. \\
&\quad \left. \left. N_1 \left(\inf_{K \in S} K(c_1) \right), \dots, N_k \left(\inf_{K \in S} K(c_k) \right) \right), \right. \\
&\quad \left. g \left(\min \left(M'(a_1), \inf_{K \in S} K(a_1) \right), \dots, \min \left(M'(a_n), \inf_{K \in S} K(a_n) \right) \right) \right) \\
&= I \left(f \left(\min \left(M'(b_1), M(b_1) \right), \dots, \min \left(M'(b_m), M(b_m) \right), \right. \right. \\
&\quad \left. \left. N_1 \left(\inf_{K \in S} K(c_1) \right), \dots, N_k \left(\inf_{K \in S} K(c_k) \right) \right), \right. \\
&\quad \left. g \left(\min \left(M'(a_1), M(a_1) \right), \dots, \min \left(M'(a_n), M(a_n) \right) \right) \right) \\
&= I \left(f \left(M'(b_1), \dots, M'(b_m), N_1 \left(\inf_{K \in S} K(c_1) \right), \dots, N_k \left(\inf_{K \in S} K(c_k) \right) \right), \right. \\
&\quad \left. g \left(M'(a_1), \dots, M'(a_n) \right) \right) \\
&= I \left(f \left(\inf_{K \in S'} K(b_1), \dots, \inf_{K \in S'} K(b_m), N_1 \left(\inf_{K \in S} K(c_1) \right), \dots, N_k \left(\inf_{K \in S} K(c_k) \right) \right), \right. \\
&\quad \left. g \left(\inf_{K \in S'} K(a_1), \dots, \inf_{K \in S'} K(a_n) \right) \right) \\
&= [f(B b_1, \dots, B b_m, \text{not}_1 c_1, \dots, \text{not}_k c_k) \rightarrow g(B a_1, \dots, B a_n)]_{(I', S', S)} \\
&= [\mu(r)]_{(I', S', S)}
\end{aligned}$$

□

Proof of Theorem 6

Theorem 8. *Consider a general FASP program P and $M \in \mathcal{F}(\mathcal{B}_P)$. M is an answer set of P iff $\text{Mod}(M)$ is a fuzzy autoepistemic model of $\sigma(P)$.*

Proof. (\Rightarrow) First suppose that M is an answer set of P . By Theorem 5, it follows that (I, S) with $S = \text{Mod}(M)$ and $I \in \mathcal{F}(A)$ arbitrary is a FMBNF model of $\mu(P)$. By Definition 8 this means that

- (a) for each $\alpha \in \mu(P)$: $[\alpha]_{(I, S, S)} = 1$ and
- (b) there is no structure (I', S') such that $S \subset S'$ and $[\alpha]_{(I', S', S)} = 1$ for all $\alpha \in \mu(P)$.

We need to prove that S is a fuzzy autoepistemic model of $\sigma(P)$, or by Definition 5 that

$$S = \{K \in \mathcal{F}(A) \mid \forall \varphi \in \sigma(P) : [\varphi]_{(K, S)} = 1\}.$$

Let $J \in S$, thus $S \cup \{J\} = S$. By Lemma 9 it follows that $[\sigma(r)]_{(J, S)} = [\mu(r)]_{(I, S, S)} = 1$ for all $r \in P$. Hence

$$J \in \{K \in \mathcal{F}(A) \mid \forall \varphi \in \sigma(P) : [\varphi]_{(K, S)} = 1\}$$

as soon as $J \in S$. Conversely, suppose that $J \in \mathcal{F}(A)$ such that $[\varphi]_{(J, S)} = 1$ for every $\varphi \in \sigma(P)$. If $J \notin S$ define $S' = S \cup \{J\}$. By Lemma 9 it follows that $[\mu(r)]_{(J, S', S)} = [\sigma(r)]_{(J, S)} = 1$ for each $r \in P$. This contradicts the maximality of S .

(\Leftarrow) Now suppose that $S = \text{Mod}(M)$ is a fuzzy autoepistemic model of $\sigma(P)$. By Theorem 5, it is sufficient to show that (I, S) with I an arbitrary fuzzy interpretation in $\mathcal{F}(A)$ is a FMBNF model of $\mu(P)$. By Definition 8 we need to show that

- (a) for each $\alpha \in \mu(P)$: $[\alpha]_{(I, S, S)} = 1$ and
- (b) there is no structure (I', S') such that $S \subset S'$ and $[\alpha]_{(I', S', S)} = 1$ for all $\alpha \in \mu(P)$.

Let $J \in S$, thus $S \cup \{J\} = S$. By Lemma 9, it follows that $[\mu(r)]_{(I, S, S)} = [\sigma(r)]_{(J, S)}$ for each $r \in P$. Since S is a fuzzy autoepistemic model of $\sigma(P)$ and thus

$$S = \{K \in \mathcal{F}(A) \mid \forall \varphi \in \sigma(P) : [\varphi]_{(K, S)} = 1\}$$

we have $[\alpha]_{(I, S, S)} = 1$ for all $\alpha \in \mu(P)$. Now suppose there is a structure (I', S') such that $S \subset S'$ and $[\alpha]_{(I', S', S)} = 1$ for all $\alpha \in \mu(P)$. Define

$$M' : \mathcal{B}_P \rightarrow [0, 1] : a \mapsto \inf_{J \in S'} J(a).$$

For $K \in S'$ it holds that $M'(a) = \inf_{J \in S'} J(a) \leq K(a)$ for all $a \in \mathcal{B}_P$, hence $S' \subseteq \text{Mod}(M')$ and thus $\text{Mod}(M) = S \subset S' \subseteq \text{Mod}(M')$. By Lemma 4 it follows that $M' < M$. Note that S' and $\text{Mod}(M')$ are inf-equivalent w.r.t. P by definition of M' . By Lemma 10 it then follows that $[\sigma(r)]_{(M',S)} = [\mu(r)]_{(I',S',S)} = 1$ which implies that $M' \in S = \text{Mod}(M)$ since S is a fuzzy autoepistemic model of $\sigma(P)$. Hence $M \leq M'$, a contradiction.

□

References

- [1] C. Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, 2003.
- [2] C. Baral, M. Gelfond, Logic programming and knowledge representation, Journal of Logic Programming 19 (1994) 73–148.
- [3] R. Moore, Semantical considerations on nonmonotonic logic, in: Proceedings of the Eight International Joint Conference on Artificial Intelligence, 1983, pp. 272–279.
- [4] R. Moore, Possible-world semantics in autoepistemic logic, in: Proceedings of the Non-Monotonic Reasoning Workshop, 1984, pp. 344–354.
- [5] M. Gelfond, On stratified autoepistemic theories, in: Proceedings of the Sixth National Conference on Artificial Intelligence, 1987, pp. 207–211.
- [6] M. Gelfond, V. Lifschitz, The Stable Model Semantics for Logic Programming, in: Proceedings of the Fifth International Conference and Symposium on Logic Programming, 1988, pp. 1070–1080.
- [7] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Generation Computing 9 (1991) 365–385.
- [8] V. Lifschitz, G. Schwarz, Extended logic programs as autoepistemic theories, in: Proceedings of the Second International Workshop on Logic Programming and Nonmonotonic Reasoning, 1993, pp. 101–114.
- [9] F. Lin, Y. Shoham, A logic of knowledge and justified assumptions, Artificial Intelligence 57 (1992) 271–289.

- [10] V. Lifschitz, Minimal belief and negation as failure, *Artificial Intelligence* 70 (1994) 53–72.
- [11] J. Chen, Minimal knowledge + negation as failure = only knowing (sometimes), in: *Proceedings of the Second International Workshop on Logic Programming and Nonmonotonic Reasoning*, 1993, pp. 132–150.
- [12] J. Halpern, Y. Moses, Towards a theory of knowledge and ignorance: Preliminary report, in: *Logic and Models of Concurrent Systems*, 1985, pp. 459–476.
- [13] J. Janssen, S. Schockaert, D. Vermeir, M. De Cock, General Fuzzy Answer Set Programs, in: *Proceedings of the International Workshop on Fuzzy Logic and Applications*, 2009, pp. 353–359.
- [14] D. Van Nieuwenborgh, M. De Cock, D. Vermeir, An introduction to fuzzy answer set programming, *Annals of Mathematics and Artificial Intelligence* 50 (3-4) (2007) 363–388.
- [15] P. Hájek, *Metamathematics of Fuzzy Logic*, Trends in Logic, 1998.
- [16] M. Blondeel, S. Schockaert, M. De Cock, D. Vermeir, Fuzzy autoepistemic logic: Reflecting about knowledge of truth degrees, in: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 2011, pp. 616–627.
- [17] G. von Wright, *An Essay in Modal Logic*, Studies in Logic and the Foundations of mathematics, Amsterdam: Nort-Holland Pub. Co., 1951.
- [18] K. Sim, Epistemic Logic and Logical Omniscience: A Survey, *International Journal of Intelligent Systems* 12 (1997) 57–81.
- [19] D. McDermott, Nonmonotonic logic 2: Nonmonotonic modal theories, *Journal of the Association for Computing Machinery* 29 (1) (1982) 33–57.
- [20] D. McDermott, J. Doyle, Nonmonotonic logic 1, *Artificial Intelligence* 13 (1-2) (1980) 41–72.
- [21] M. Banerjee, D. Dubois, A Simple Modal Logic for Reasoning about Revealed Beliefs, in: *Proceedings of the 10th European Conference on*

Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 2009, pp. 805–816.

- [22] W. Marek, M. Truszczyński, Stable semantics for logic programs and default reasoning, in: Proceedings of the North American Conference on Logic Programming, 1989, pp. 243–257.
- [23] V. Lifschitz, Between circumscription and autoepistemic logic, in: Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, 1989, pp. 235–244.
- [24] G. Schwarz, Autoepistemic logic of knowledge, in: Logic Programming and Nonmonotonic Reasoning: Proceedings of the First International Workshop, 1991, pp. 260–274.
- [25] F. Lin, Y. Shoham, Argument systems: a uniform basis for nonmonotonic reasoning, in: Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, 1989, pp. 245–255.
- [26] G. Schwarz, Reflexive autoepistemic logic, *Fundamenta Informaticae* 17 (1992) 157–173.
- [27] T. Łukasiewicz, U. Straccia, Tightly integrated fuzzy description logic programs under the answer set semantics for the semantic web, in: Proceedings of the 1st International Conference on Web Reasoning and Rule Systems, 2007, pp. 289–298.
- [28] C. Damásio, L. Pereira, Antitonic Logic Programs, in: Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning, 2001, pp. 379–392.
- [29] M. Fitting, Fixpoint semantics for logic programming: a survey, *Theoretical Computer Science* 278 (1-2) (2002) 25–51.
- [30] U. Straccia, Annotated answer set programming, in: Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2006.
- [31] Y. Loyer, U. Straccia, Epistemic foundation of stable model semantics, *Theory and Practice of Logic Programming* 6 (4) (2006) 355–393.

- [32] U. Straccia, Query answering in normal logic programs under uncertainty, in: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 2005, pp. 687–700.
- [33] U. Straccia, Query answering under the any-world assumption for normal logic programs, in: *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning*, 2006, pp. 329–339.
- [34] S. Schockaert, J. Janssen, D. Vermeir, M. De Cock, Answer sets in a fuzzy equilibrium logic, in: *Proceedings of the 3rd International Conference in Web Reasoning and Rule Systems*, 2009, pp. 135–149.
- [35] U. Straccia, M. Ojeda-Aciego, C. V. Damásio, On fixed-points of multi-valued functions on complete lattices and their application to generalized logic programs, *SIAM Journal on Computing* (5) (2009) 1881–1911.
- [36] D. Dubois, H. Prade, Possibility theory, probability theory and multiple-valued logics: a clarification, *Annals of Mathematics and Artificial Intelligence* 32 (1-4) (2001) 35–66.
- [37] P. Nicolas, L. Garcia, I. Stéphan, C. Lefèvre, Possibilistic uncertainty handling for answer set programming, *Annals of Mathematics and Artificial Intelligence* 47 (1-2) (2006) 139–181.
- [38] C. Baral, M. Gelfond, J. Rushton, Probabilistic reasoning in computer science, in: *Logic Programming and Nonmonotonic Reasoning*, 7th International Conference, 2004, pp. 21–33.
- [39] P. Hájek, On fuzzy modal logics, *Fuzzy Sets and Systems* 161 (18) (2010) 2389–2396.
- [40] A. Mironov, Fuzzy Modal Logics, *Journal of Mathematical Sciences* 128 (6) (2005) 3461–3483.
- [41] M. Fitting, Many-valued modal logics, in: *Fundamenta Informaticae*, 1992, pp. 365–448.
- [42] M. Fitting, Many-valued modal logics II, *Fundamenta Informaticae* 17.
- [43] M. Fitting, Many-valued non-monotonic modal logics, in: *Logic Foundations of Computer Science*, 1992, pp. 139–150.

- [44] C. Koutras, S. Zachos, Many-Valued Reflexive Autoepistemic Logic, *Logic Journal of the IGPL* 8 (1) (2000) 403–418.
- [45] M. Van Emden, R. Kowalski, The semantics of predicate logic as a programming language, *Journal of the ACM* 23 (1976) 569–574.
- [46] R. Stalnaker, A note on non-monotonic modal logic, *Artificial Intelligence* 64 (2) (1993) 183–196.
- [47] W. Marek, Stable theories in autoepistemic logic, unpublished note, Department of Computer Science, University of Kentucky (1986).
- [48] R. McNaughton, A theorem about infinite-valued sentential logic, *The Journal of Symbolic Logic* 16 (1).
- [49] R. Hähnle, Proof theory of many-valued logic - linear optimization - logic design: connections and interactions, *Soft Computing* 1 (1997) 107–119.
- [50] T. Alsinet, L. Godo, A complete calculus for possibilistic logic programming with fuzzy propositional variables, in: *Uncertainty in Artificial Intelligence*, 2000, pp. 1–10.
- [51] L. Yu, N. Wang, X. Meng, Real-time forest fire detection with wireless sensor networks, in: *Proceedings of the Wireless Communication Networking and Mobile Computing International Conference*, 2005, pp. 1214–1217.
- [52] H. Levesque, Foundations of a functional approach to knowledge representation, *Artificial Intelligence* 23 (2) (1984) 155–212.
- [53] R. Reiter, What should a database know?, *Journal of Logic Programming* 14 (1991) 127–153.
- [54] U. Egly, T. Eiter, H. Tompits, S. Woltran, Solving Advanced Reasoning Tasks Using Quantified Boolean Formulas, in: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000, pp. 417–422.
- [55] M. Blondeel, S. Schockaert, M. De Cock, D. Vermeir, NP-completeness of fuzzy answer set programming under Łukasiewicz semantics, submitted.